Computational Adequacy for Recursive Types in Models of Intuitionistic Set Theory

Simpson, A. (2004). Computational Adequacy for Recursive Types in Models of Intuitionistic Set Theory. Annals of Pure and Applied Logic, 130(1-3), 207-275doi: 10.1016/j.apal.2003.12.005

Computational Adequacy for Recursive Types in Models of Intuitionistic Set Theory

Alex Simpson [1]

LFCS, School of Informatics, University of Edinburgh, Scotland

Abstract

This paper provides a unifying axiomatic account of the interpretation of recursive types that incorporates both domain-theoretic and realizability models as concrete instances. Our approach is to view such models as full subcategories of categorical models of intuitionistic set theory. It is shown that the existence of solutions to recursive domain equations depends upon the strength of the set theory. We observe that the internal set theory of an elementary topos is not strong enough to guarantee their existence. In contrast, as our first main result, we establish that solutions to recursive domain equations do exist when the category of sets is a model of full intuitionistic Zermelo-Fraenkel set theory. We then apply this result to obtain a denotational interpretation of FPC, a recursively typed lambda-calculus with call-by-value operational semantics. By exploiting the intuitionistic logic of the ambient model of intuitionistic set theory, we analyse the relationship between operational and denotational semantics. We first prove an "internal" computational adequacy theorem: the model always believes that the operational and denotational notions of termination agree. This allows us to identify, as our second main result, a necessary and sufficient condition for genuine "external" computational adequacy to hold, i.e. for the operational and denotational notions of termination to coincide in the real world. The condition is formulated as a simple property of the internal logic, related to the logical notion of 1-consistency. We provide useful sufficient conditions for establishing that the logical property holds in practice. Finally, we outline how the methods of the paper may be applied to concrete models of FPC. In doing so, we obtain computational adequacy results for an extensive range of realizability and domain-theoretic models.

Key words: Domain theory, Algebraic compactness, FPC
2000 MSC: 03B70, 03C90, 03E70, 03E73, 03D45, 03F55, 18B25, 68Q55

[1] Research supported by EPSRC Research Grant no. K06109 (1998–2002), an EPSRC Advanced Research Fellowship (2001–), and a visiting professorship at RIMS, Kyoto University (2002–2003).

1 Introduction

In his work on *algebraic compactness*, Freyd [9,10] identified the categorical structure required to model recursive types. Many examples of algebraically compact categories are known. Domain theory provides the classical example of the category of ωcpos [3]. More generally, axiomatic domain theory has successfully abstracted the particularities of domains to provide a host of "neo-classical" models [3,6]. A very different type of model is given by game-theoretic semantics [25]. Finally, there are a variety of models based on realizability [11,28–30,21,22,35]. What has been missing hitherto is a single unifying treatment accounting for the existence of all these types of model. In this paper, we provide the axiomatic basis for such a treatment. In a follow-up paper [44], we shall demonstrate how the various types of model are incorporated within our axiomatic framework.

Categories that model recursive types have nontrivial fixed-point operators and thus, by a simple argument using classical logic, cannot be full subcategories of the category of sets. In [38], Dana Scott showed that such categories can nonetheless live as full subcategories of models of *intuitionistic* set theory, an observation that led to the subsequent development of *synthetic domain theory* [36,14,28,46,22,40,35,27,7]. In this paper, we exploit this idea to obtain algebraically compact categories in a uniform way. Roughly speaking, we start off with a category $\mathbf{S}$ of intuitionistic sets that satisfies one simple axiom, Axiom **1** of Section 2. From any such category $\mathbf{S}$, we extract a full subcategory of *predomains*, $\mathbf{P} \hookrightarrow \mathbf{S}$, whose associated category of partial maps, $\mathbf{pP}$, is algebraically compact.

This approach directly follows [40], where it is shown that a model of the simply-typed language PCF [32] can be similarly extracted from any elementary topos $\mathbf{S}$, with natural numbers object, satisfying a stronger Axiom **N**. The additional goal of the present paper is to show the algebraic compactness of $\mathbf{pP}$. This is a nontrivial task.

In fact, we immediately encounter a problem. As our first result, Proposition 2.7, we show that there exists an elementary topos satisfying Axiom **N** for which the derived category $\mathbf{pP}$ is *not* algebraically compact. Thus some modification to the above method of constructing $\mathbf{P}$ is necessary in order to interpret recursive types. This is not, at first sight, surprising. Axiom **N** is designed merely to guarantee that $\mathbf{P}$ models the recursive definition of functions. Thus there is no *a priori* reason to expect recursive types to have interpretations in $\mathbf{pP}$.

However, we identify the difficulty as stemming from a perhaps unexpected source. The problem is that elementary toposes, although models of intuition-

istic higher-order logic, are not, in general, models of a sufficiently powerful set theory. Thus, instead of working with an arbitrary elementary topos, we shall require that **S** have enough structure to model full *Intuitionistic Zermelo-Fraenkel (IZF)* set theory, see e.g. [37]. Technically, this is implemented by asking for **S** to be given as the full subcategory of *small* objects in a category **C** with *class(ic) structure* and *universal object*, in the sense of [41,43] (developed from [19]). As our first main result, Theorem 1, we prove that, with such a category **S**, the derived category **pP** is algebraically compact whenever Axiom **1** holds. Thus, with enough set-theoretic power to back it up, Axiom **1** is, after all, sufficient for the solution of recursive domain equations.

As an application of Theorem 1, we give an interpretation of FPC, a recursively-typed λ-calculus with call-by-value operational semantics, in **pP**. As our second main result, Theorem 2, we obtain necessary and sufficient conditions for this interpretation to be *computationally adequate*, i.e. for the denotational and operational notions of program termination to agree. Our approach to computational adequacy makes extensive use of the intuitionistic set theory of the ambient category **C**. The method of attack is to first prove an "internal" computational adequacy result: the model always believes itself to be computationally adequate, Proposition 12.2. Next we identify a logical property that characterizes the circumstances in which the "internal" belief of computational adequacy coincides with "external" reality. The logical property is a simple one, closely related to the proof-theoretic notion of 1-*consistency* [12, Def. 1.3.6]. We also provide useful sufficient conditions for verifying that the logical property holds in models that occur in practice.

Our approach to computational adequacy is based on a similar characterization of computational adequacy for the simply-typed language PCF in [40]. However, the present paper extends the results of *op. cit.* in two significant ways. Firstly, we assume the weaker Axiom **2** rather than Axiom **N**, see Section 2. This is important in the applications of our results in Section 15. Second, the extension of the proof of "internal" computational adequacy to include recursive types involves a substantial amount of nontrivial work.

The paper naturally divides into two halves. The first half entirely concerns algebraic compactness, and is centred around Theorem 1. In Sections 2 and 3 we present the necessary background material to formulate Theorem 1. The proof of the theorem then occupies a large chunk of the paper, Sections 4–9. To help orientate the reader, a detailed outline of the proof structure is given in Section 3.

In the second half of the paper, we address the question of computational adequacy for FPC. Section 10 presents a brief overview of the FPC language and its operational semantics. In Section 11, we apply Theorem 1 to obtain a denotational interpretation of FPC in **pP**. The proof of "internal" computa-

tional adequacy is given in Sections 12 and 13. Finally, our main computational adequacy result, Theorem 2, is stated and proved in Section 14.

The main applications of the results of this paper are to establish computational adequacy for axiomatically given classes of models of FPC, including domain-theoretic and realizability models. In Section 15, we briefly outline our results in this direction. Full details will appear in a follow-up paper [44].

The research in the present paper constitutes a development of the techniques of *synthetic domain theory*. Nevertheless, our applications to axiomatically given classes of models demonstrate that our results should be viewed equally much as a contribution to the field of *axiomatic domain theory* [9,10,39,4,3,5,6]. It is the author's view that embedding categories of predomains within models of intuitionistic set theory is the correct approach to obtaining an axiomatic account of domain-theoretic constructions that applies uniformly across the different types of model. At present, it is the only known approach.

2 Classes, sets and predomains

As discussed in the introduction, our work will involve both elementary toposes and also categorical models of Intuitionistic Zermelo-Fraenkel (IZF) set theory [37]. Both types of model arise as instances of regular categories with *class(ic) structure*, as defined in [41,43]. We briefly recount the main features of this notion, using, as far as possible, set-theoretic intuition. For the category-theoretic details see *op. cit.*

A category is *regular* if: it has finite limits, every morphism $f\colon X \longrightarrow Y$ factorizes as

$$f \;=\; X \xrightarrow{\;e\;} I \xrightarrow{\;m\;} Y,$$

where e is regular epi and m is mono, and such factorizations are stable under pullback. Intuitively, I represents the image of f in Y, the regular epi e is the surjection onto the image, and the mono m is the inclusion of the image in Y. *Class structure* is additional structure on a regular category, designed to ensure that the objects of the category behave like classes, and that the morphisms behave like functions between classes. In particular, in any regular category $\mathbf{C}$ with class structure, there is a distinguished full subcategory, $\mathbf{S}$, of *small* objects, which is to be thought of as the subcategory of sets. More generally, there is a distinguished collection of morphisms, the *small maps*, where intuitively $f\colon X \longrightarrow Y$ is small if, for every y in the class Y, its fibre $f^{-1}(y)$, which is a subclass of the class X, is actually a set. Smallness interacts with the regular structure on $\mathbf{C}$ as follows. If $X \rightarrowtail Y$ is mono and Y is small then X is small, i.e. every subclass of a set is a set. This expresses the Separation axiom of set theory. Dually, if $X \twoheadrightarrow Y$ is regular epi and X is

small then Y is small, i.e. the image of a function from a set to a class is itself a set. This expresses the Replacement axiom of set theory. The other important structure on $\mathbf{C}$ is that, for every class X, there is another class $\mathcal{P}_{\mathbf{S}}X$ the *small powerobject* of X, which is intuitively the class of all subsets of X. The object $\mathcal{P}_{\mathbf{S}}X$ comes with an associated *membership* relation $\ni_X \rightarrowtail \mathcal{P}_{\mathbf{S}}X \times X$, for which the composite

$$\gamma_X \quad = \quad \ni_X \rightarrowtail \mathcal{P}_{\mathbf{S}}X \times X \xrightarrow{\pi_1} \mathcal{P}_{\mathbf{S}}X \tag{1}$$

is a small map. It is also required that if X is small then so is $\mathcal{P}_{\mathbf{S}}X$. This expresses the Powerset axiom of set theory. It follows that the full subcategory $\mathbf{S}$ is an elementary topos. Further, $\mathbf{C}$ has finite coproducts and $\mathbf{S}$ is closed under finite limits and coproducts in $\mathbf{C}$. Also, every epi in $\mathbf{C}$ is regular.

Importantly, $\mathbf{C}$ supports an internal logic, which is intuitionistic first-order logic. Intuitively, one thinks of the subobjects of an object X as being predicates on X. The lattice of subobjects, $\mathrm{Sub}(X)$ is a Heyting algebra. Moreover, for any $f\colon X \longrightarrow Y$, the inverse image map $f^{-1}\colon \mathrm{Sub}(Y) \longrightarrow \mathrm{Sub}(X)$ has left and right adjoints, giving existential and universal quantification respectively; see [41,43] for details. We shall make liberal use of this internal logic, writing $\mathbf{C} \models \varphi$ to mean that statement φ holds internally in $\mathbf{C}$. The object $\Omega = \mathcal{P}_{\mathbf{S}}\mathbf{1}$ (where $\mathbf{1}$ is the terminal object in $\mathbf{C}$), which is the subobject classifier in $\mathbf{S}$, is also a subobject classifier in $\mathbf{C}$. Thus Ω can be thought of as the set of all internal propositions in $\mathbf{C}$.

As we shall make heavy use of indexed families in $\mathbf{C}$, we summarise the legitimate constructions on them in the context of class structure. As usual, we consider I-indexed families as being given by morphisms $X \longrightarrow I$, although we shall often use the convenient notation $\{X_i\}_{i:I}$ for them. Given such an internal family $X \longrightarrow I$, the object X itself provides a dependent sum $\sum_{i:I} X_i$. However, a dependent product $\prod_{i:I} X_i$ is only guaranteed to exist in the case that I is a small object. If, in addition to I being small, $X \longrightarrow I$ is a small map then $\prod_{i:I} X_i$ is itself a small object. In the case of a constant family $\{X\}_{y:Y}$ (given by a projection $X \times Y \longrightarrow Y$), dependent product specialises to function space. Thus the above remarks imply that the internal function space Y^X (for which we also write $X \to Y$) exists whenever X is a small object, and that Y^X is itself small if both X and Y are small.

An important fact about the class structure is that it is preserved by slicing, i.e. for every object I of $\mathbf{C}$, the slice category $\mathbf{C}/I$ also has class structure, see [41, Theorem 2]. In practical terms, this means we can often derive a parametrized version of a result, with a free parameter in I, from an unparametrized external result about $\mathbf{C}$.

Henceforth in this paper, let $\mathbf{C}$ be a regular category with class structure, and let $\mathbf{S}$ be its full subcategory of small objects. We also assume that $\mathbf{C}$

has a small natural numbers object (nno) $\mathbf{N}$. This implements the Infinity axiom of set theory. Further, we assume that all the assumed structure is specified, i.e. we have chosen finite limits, chosen image factorizations, etc. Henceforth whenever we impose additional structure on $\mathbf{C}$ we again assume, without further comment, that the structure is specified.

In spite of the motivating references to set theory, the assumed structure on $\mathbf{C}$ and $\mathbf{S}$ does not yet provide the full power of IZF set theory. For example, given any elementary topos with nno, $\mathbf{S}$, one can obtain class structure by putting $\mathbf{C} = \mathbf{S}$ and stipulating that every map be small. At present, the two-tiered structure serves only to allow the topos $\mathbf{S}$ to live within the environs of a possibly larger surrounding universe of classes. This facility will play a crucial rôle from Section 3 onwards, but it is introduced at this stage merely to permit a unified presentation of the material.

Before starting on the technical work proper, we summarize a few stylistic and notational conventions that we use. The results in the paper include *external* results about the category $\mathbf{C}$ (and derivatives of it) and *internal* results derived within the internal logic of $\mathbf{C}$. We try to be as clear as possible about where the divisions between external and internal reasoning lie, but at the same time we try to avoid being overly pedantic. The problem of maintaining a rigorous, though readable, separation between internal and external reasoning is a thorny one, especially in a paper of this length. It is to be hoped that the following conventions are sufficient to help the reader through the paper. When reasoning internally, we write $x : X$ in order to identify an object X as the type of an expression x. On the other hand, we write $x \in X'$ if X' is understood as being a subobject $X' \rightarrowtail X$ where $x : X$ and x lies within the subobject. We also write $x \in y$ if $x : X$ and $y : \mathcal{P}_{\mathbf{S}} X$ and (y, x) lies in the subobject $\ni_X \rightarrowtail \mathcal{P}_{\mathbf{S}} X \times X$. We make various notational distinctions between external structure and internal structure, e.g. $\mathbb{L}$ is an external functor on $\mathbf{C}$, whereas, in Section 9, L is the corresponding internal functor on an internal category. Similarly, we write $f : X \longrightarrow Y$ for a morphism in the category $\mathbf{C}$, and we write $f : Y^X$ (and $f : X \to Y$) to type f internally as an element of the object Y^X. However, the reader should be warned that, we use these conventions quite flexibly. In particular, we often, e.g., work with some assumed $i : I$ and then continue to use the notation for external structure. Formally, this should be understood as moving to the slice category $\mathbf{C}/I$ and referring to external structure on that category.

The remaining goal of this section is to isolate a full subcategory of $\mathbf{S}$ to act as a category of predomains. This will require imposing further axioms on $\mathbf{C}$. Many axiomatizations have been suggested for this purpose, see e.g. [36,14,28,46,22,40,35,27]. Here, we follow [22,40].

As first proposed in [36], the definition of predomain is predicated on a no-

tion of partiality. To implement this, we require a distinguished subobject $\Sigma \rightarrowtail \Omega$. As Σ is a subobject of Ω, it classifies a collection of subobjects in $\mathbf{C}$, namely those whose characteristic map to Ω factors through $\Sigma \rightarrowtail \Omega$. We call such subobjects Σ-*subobjects*, and we write $(X \upharpoonright p)$ for the Σ-subobject $(X \upharpoonright p) \rightarrowtail X$ determined by $p\colon X \longrightarrow \Sigma$.

Intuitively, the object Σ is intended to correspond to the subobject of those propositions in Ω that express the termination of programs. Because there exist terminating programs, and because programs can be run under sequential composition, it makes sense to require that Σ contains the true proposition, $\top$, and that Σ-subobjects are closed under composition. This implies, in particular, that Σ is closed under finite conjunction in Ω. Taken together, these requirements state that Σ is a *dominance* in the sense of [36].

The dominance Σ determines a *lifting* functor on $\mathbf{C}$. For an object X, we say (internally in $\mathbf{C}$) that $e\colon \mathcal{P}_{\mathbf{S}}X$ is *subterminal* if

$$\forall x, x'\colon X.\ x \in e \wedge x' \in e \rightarrow x = x'.$$

We say that e is Σ-*subterminal* if it is subterminal and also

$$(\exists x\colon X.\ x \in e) \in \Sigma,$$

i.e. the proposition stating that e is inhabited is a Σ-proposition. Using the internal logic of $\mathbf{C}$, define

$$\mathbb{L}X = \{e\colon \mathcal{P}_{\mathbf{S}}X \mid e \text{ is } \Sigma\text{-subterminal}\}.$$

The $\mathbb{L}$ operation extends to a functor $\mathbb{L} : \mathbf{C} \rightarrow \mathbf{C}$, where, on $f\colon X \longrightarrow Y$, the morphism action $\mathbb{L}f\colon \mathbb{L}X \longrightarrow \mathbb{L}Y$ is defined by

$$(\mathbb{L}f)(e) = \{f(x) \mid x \in e\}.$$

Further, the endofuctor $\mathbb{L}$ carries a monad structure. The unit is singleton $\{\cdot\} : X \longrightarrow \mathbb{L}X$, and the multiplication is union $\bigcup : \mathbb{L}\mathbb{L}X \longrightarrow \mathbb{L}X$.

As in [18], the endofunctor $\mathbb{L}$ has a final coalgebra, $\tau\colon \mathbf{F} \longrightarrow \mathbb{L}\mathbf{F}$ (necessarily an isomorphism), defined by:

$$\begin{aligned} \mathbf{F} &= \{c\colon \Sigma^{\mathbf{N}} \mid \forall n\colon \mathbf{N}.\ c(n+1) \rightarrow c(n)\} \\ \tau(c) &= \{(n \mapsto c(n+1)) \mid c(0)\}. \end{aligned}$$

Because $\mathbf{F}$ is small and the $\mathbb{L}$ functor preserves subset inclusions, there exists a smallest subalgebra, $\sigma : \mathbb{L}\mathbf{I} \longrightarrow \mathbf{I}$, of τ^{-1}, defined internally in $\mathbf{C}$ as the intersection of all subalgebras of τ^{-1}. It is a consequence of [41, Theorem 5] that $\sigma : \mathbb{L}\mathbf{I} \longrightarrow \mathbf{I}$ is an initial algebra for the endofunctor $\mathbb{L}$ on $\mathbf{C}$. By construction, the unique algebra homomorphism, $\iota : \mathbf{I} \longrightarrow \mathbf{F}$, from σ to τ^{-1}

is mono. It is not epi (unless $\mathbf{C}$ is trivial), because $\infty = (n \mapsto \top)$ is a point in $\mathbf{F}$ that is not in the image of ι. Henceforth, we shall ignore the explicit constructions of $\mathbf{I}$ and $\mathbf{F}$ given above, and instead work purely with their universal properties as the initial $\mathbb{L}$-algebra and final $\mathbb{L}$-coalgebra respectively.

Informally, one can view $\mathbf{I}$ as the object obtained from the initial object $\mathbf{0}$ by freely iterating the $\mathbb{L}$ functor. In the sequel, $\mathbf{I}$ will play the rôle of a generic "ω-chain" in $\mathbf{C}$, and $\mathbf{I} \rightarrowtail \mathbf{F}$ will exhibit $\mathbf{F}$ as its "chain-completion". This intuition plays a fundamental rôle in developing a basic notion of "chain completeness" used to define a full subcategory of predomains within $\mathbf{S}$, see [22]. On the other hand, one must be careful with this intuition, as we do not (yet) have axioms that ensure that $\mathbf{I}$ is even inhabited.

Proposition 2.1 *For any object X, the following are equivalent.*

(1) The map $X^{\iota} : X^{\mathbf{F}} \longrightarrow X^{\mathbf{I}}$ is an isomorphism.
(2) $\mathbf{C} \models \forall f : X^{\mathbf{I}}.\, \exists! f' : X^{\mathbf{F}}.\, \forall i : \mathbf{I}.\, f(i) = f'(\iota(i))$.

Definition 2.2 (Complete object) An object X is said to be *complete* if either of the equivalent conditions of Proposition 2.1 hold.

Examples in [27] show that complete objects do not themselves form a suitable category of predomains as they are not necessarily closed under lifting. Following [22], we avoid this problem using the property of *well-completeness*.

Definition 2.3 (Well-complete object) An object X is *well-complete* if $\mathbb{L}X$ is complete.

The results below, which are standard, state the basic properties of well-completeness. In them, we write $\mathbf{2}$ for the object $\mathbf{1} + \mathbf{1}$, which we view as a subobject of Ω via $[\bot, \top] : \mathbf{2} \rightarrowtail \Omega$, where $\bot$ is *falsum*. We sometimes refer to $\mathbf{2}$ as the object of *logically decidable* propositions, because for any proposition p we have $p \in \mathbf{2}$ if and only if $\mathbf{C} \models p \vee \neg p$.

Proposition 2.4

(1) If $\mathbf{2}$ is well-complete then so are $\mathbf{1}$ and $\mathbf{0}$.
(2) If $\mathbf{N}$ is well-complete then so is $\mathbf{2}$.

The converse implications do not hold in general, see [27].

Proposition 2.5 *If $\mathbf{1}$ is well-complete then:*

(1) X is well-complete if and only if

$$\mathbf{C} \models \forall p : \Sigma^{\mathbf{F}}.\, \forall f : X^{(\mathbf{I} \restriction p \circ \iota)}.\, \exists!\, f' : X^{(\mathbf{F} \restriction p)}.\, \forall i : (\mathbf{I} \restriction p \circ \iota).\, f'(\iota(i)) = f(i).$$

(2) X *well-complete implies* X *complete.*
(3) X *well-complete implies* $\mathbb{L}X$ *well-complete.*
(4) For any internal family $\{X_i\}_{i:\,I}$ *with* I *small,*

$$\mathbf{C} \models (\forall i : I.\ X_i \text{ is well-complete}) \rightarrow (\prod_{i:\,I} X_i) \text{ is well-complete.}$$

Two special cases:
If X, Y *are well-complete then so is* $X \times Y$.
If X *is small and* Y *is well-complete then* Y^X *is well-complete.*
(5) Given two morphisms $f, g\colon X \longrightarrow Y$ *with* X, Y *well-complete then, for any equalizer* $e\colon E \rightarrowtail X$ *of* f *and* g*, the object* E *is well-complete.*
(6) For any internal family $\{X_i \rightarrowtail X\}_{i:\,I}$ *of subobjects of a well-complete object* X*, let* $(\bigcap_{i:\,I} X_i) \rightarrowtail X$ *be the intersection. Then:*

$$\mathbf{C} \models (\forall i : I.\ X_i \text{ well-complete}) \rightarrow (\bigcap_{i:\,I} X_i) \text{ well-complete.}$$

(7) Given a subobject $X' \rightarrowtail X$ *and* $f\colon Y \longrightarrow X$ *where* X*,* X' *and* Y *are all well-complete, then* $f^{-1}X'$ *is well-complete, where*

$$f^{-1}X' = \{y : Y \mid f(y) \in X'\}.$$

(8) $\mathbf{0}$ *is well-complete if and only if* $\bot \in \Sigma$.
(9) $\mathbf{2}$ *is well-complete if and only if* X, Y *well-complete implies* $X + Y$ *well-complete.*
(10) $\mathbf{N}$ *is well-complete if and only if* $\mathbf{2}$ *is well-complete and also*

$$\mathbf{C} \models \forall P : \mathbf{2}^{\mathbf{N}}.\ ((\exists n : \mathbf{N}.\ P(n)) \in \Sigma).$$

The proofs of statements (1)–(5) and (9) are routine, and essentially contained in, e.g., [22,35,7]. Statements (4) and (6) make use of the fact that well-completeness can be expressed in the internal logic of $\mathbf{C}$, e.g. using (1). Statements (6) and (7) follow easily from (4) and (5). Proofs of (8) and (10) are given in [40].

Definition 2.6 (Predomain) A *predomain* is a small well-complete object.

We write $\mathbf{P}$ for the full subcategory of predomains. Thus we have full subcategory inclusions $\mathbf{P} \hookrightarrow \mathbf{S} \hookrightarrow \mathbf{C}$. For $\mathbf{P}$ to be well behaved, we need axioms to ensure that basic objects are predomains. As all the objects we consider for this purpose are already small, the axioms are formulated in terms of well-completeness alone. We use a single format for all axioms.

Axiom X The object $\mathbf{X}$ is well-complete.

In this paper, we shall instantiate this format in three instances only: Axiom **1**, which, by Proposition 2.5(4), implies that $\mathbf{P}$ is cartesian closed; Axiom **2** which, by Proposition 2.5(9), implies that, $\mathbf{P}$ has finite coproducts (inherited

from **C**); and Axiom **N** which implies that **P** has all the structure required by a model of PCF, see [40]. The implications between these three axioms are given by Proposition 2.4.

It is worth remarking that one consequence of Axiom **1** is that the implication $(\forall p\colon \Omega.\, p \vee \neg p) \;\rightarrow\; \Sigma = \{\top\}$ holds in **C**. Axiom **1** is, in fact, consistent with **C** being a model of classical set theory, but only if the dominance is trivial. Thus, if Axiom **1** holds and $\bot \in \Sigma$ then the internal logic of **C** has to be non-classical (if it is consistent). Other consequences of Axiom **1** are: Σ is well-complete, by Proposition 2.5(3) because $\Sigma \cong \mathbb{L}\mathbf{1}$; and any Σ-subobject of a well-complete object is well-complete, by Proposition 2.5(5).

Our goal, in this paper, is to address the interpretation of recursive types in **P**. This requires that recursive domain equations have solutions up to isomorphism in an associated category **pP** of partial maps, which we now define.

For objects X, Y of **C**, a Σ-*partial map* is a partial map from X to Y whose domain $X' \rightarrowtail X$ is a Σ-subobject of X. Because Σ is a dominance, Σ-partial maps are closed under composition. As the only partial maps we are interested in are Σ-partial, we henceforth drop the Σ. We write **pC** for the category of partial maps between objects of **C**, and we write **pP** for the full subcateory of **pC** on predomains. We write $f\colon X \longrightarrow\!\!\!\!\!\rightharpoonup Y$ for a partial map from X to Y.

For later convenience, it is useful to establish notation for dealing with possibly undefined mathematical expressions resulting from the application of partial maps. We write $e\!\downarrow$ to mean that such an expression is defined. We use equality, $=$, between possibly undefined expressions for *strict equality*, i.e. $e = e'$ means that both e and e' are defined and they are equal. We also write $\simeq$ for *Kleene equality*, i.e. $e \simeq e'$ means that whenever either of e or e' is defined then so is the other and $e = e'$. We write $X \rightharpoonup Y$ for the object of partial maps from X to Y, which is easily defined in the internal logic. The object $X \rightharpoonup Y$ is isomorphic to the exponential $(\mathbb{L}Y)^X$. Thus, by Proposition 2.5(4), if Axiom **1** holds then, for X small and Y a predomain, $X \rightharpoonup Y$ is a predomain.

The first new result of this paper shows that, in the context of the assumed structure on **C**, Axiom **N** is not sufficient to allow recursive domain equations to be solved in **pP**. The statement makes use of the fact, already discussed, that any elementary topos **S** arises as the full subcategory of small objects in a category with class structure, by taking $\mathbf{C} = \mathbf{S}$.

Proposition 2.7 *There is an elementary topos satisfying Axiom* **N** *in which there exists a predomain* Υ *such that no solution* X *to the isomorphism* $X \cong X \rightharpoonup \Upsilon$ *exists in* **pP**.

PROOF (Outline). Let $\overline{\omega}$ be the set of ordinals $\leq \omega$, with their usual order-

ing endowed with the Scott topology. The Grothendieck topos $\mathcal{H}$, from [7], is the topos of sheaves over the canonical Grothendieck topology on the monoid of continuous endofunctions on $\overline{\omega}$. Let $\mathcal{H}_{\beth_\omega}$ be the full subcategory of $\mathcal{H}$ on those sheaves A for which the set $A(\overline{\omega})$ has cardinality strictly less than $\beth_\omega$ (where $\beth_0 = \aleph_0$, $\beth_{i+1} = 2^{\beth_i}$ and $\beth_\omega = \sup_{i<\omega} \beth_i$). This is an elementary topos with nno. As in [7], the category $\omega\mathbf{cpo}_{\beth_\omega}$ of ω-cpos of cardinality $< \beth_\omega$ fully embeds in $\mathcal{H}_{\beth_\omega}$ by a functor $\mathbf{y} : \omega\mathbf{cpo}_{\beth_\omega} \to \mathcal{H}_{\beth_\omega}$. Define $\Sigma = \mathbf{y}(\mathbb{S})$, where $\mathbb{S}$ is Sierpinski space. Then, as in [7], Axiom **N** is satisfied. Moreover, writing $\mathbf{P}_{\beth_\omega}$ for the full subcategory of predomains in $\mathcal{H}_{\beth_\omega}$, we have that $\mathbf{y}$ factors through the inclusion $\mathbf{P}_{\beth_\omega} \to \mathcal{H}_{\beth_\omega}$ (cf. [7]). Finally, define $\Upsilon = \mathbf{y}(Z)$ where Z is the ωcpo (the well-known countably-based L-domain that is not bifinite) drawn in [47, Example 9.6.15(c)]. The Σ-partial-function space $\rightharpoonup$ coincides with the ω-continuous partial-function space on objects of $\omega\mathbf{cpo}_{\beth_\omega}$ in $\mathbf{P}_{\beth_\omega}$. Now suppose that a solution X to $X \cong X \rightharpoonup \Upsilon$ exists in $\mathbf{pP}_{\beth_\omega}$. Let $D_0 = \mathbf{y}(\mathbf{0})$, where $\mathbf{0}$ is the empty ω-cpo. Then D_0 is a zero object in $\mathbf{pP}_{\beth_\omega}$. Define a sequence of predomains by $D_{i+1} = D_i \rightharpoonup \Upsilon$. Then, by induction, D_i is a retract of D_{i+1} in $\mathbf{P}_{\beth_\omega}$ and each D_i is also a retract of X (using the iso $X \cong X \rightharpoonup \Upsilon$). However, analogously to [47, Example 9.6.15(c)]), for each D_i, the set $D_{i+2}(\overline{\omega})$ has cardinality at least $\beth_i$. As there is a monomorphism from each D_i to X, the set $X(\overline{\omega})$ must have cardinality at least $\beth_\omega$, but this contradicts X being an object of $\mathcal{H}_{\beth_\omega}$. Thus no such object X exists. □

The above counterexample is not as strong as one would like, as one can in fact show that every closed FPC type (see Section 10) does have a solution in $\mathbf{pP}_{\beth_\omega}$. Indeed, $\mathbf{P}_{\beth_\omega}$ includes the category of countably-based bifinite domains. What the example above does show, is that the open type $\mu X.\,(X \rightharpoonup Y)$, cannot be interpreted parametrically in Y, and hence does not give an endofuctor on $\mathbf{pP}_{\beth_\omega}$. This suggests that one might, in general, restrict the notion of predomain to give a smaller better behaved category. However, this does not appear to be the way to proceed at the level of generality we are working at. Indeed, we conjecture that there exist elementary toposes satisfying Axiom **N**, for which even closed FPC types have no interpretation in **pP**.

3 Algebraic compactness

As indicated in the introduction, we address the interpretation of recursive types by strengthening the assumptions on our ambient category of classes **C**. A *universal object* is an object U such that, for every object X, there exists a mono $X \rightarrowtail U$. Thus U can be thought of as an object that collects the elements of all classes together within one universal class. In set-theoretic terms, U is simply the class of all sets (and atoms if permitted). In [41,43] it

is shown how the existence of a universal object implies that $\mathbf{C}$ contains an internal model of IZF set theory (with Replacement rather than Collection).

Henceforth we require that $\mathbf{C}$ have a universal object. For the purposes of this paper, a vital consequence of the universal object is that the categories $\mathbf{S}$, $\mathbf{P}$ and $\mathbf{pP}$ all live as internal categories within $\mathbf{C}$. In fact, the need to obtain $\mathbf{S}$ as an internal category was one of the main motivations for the development of [41,43]. Technically, this is achieved by constructing a generic small map in $\mathbf{C}$, see below. Such a generic map need not exist in the categories of classes axiomatized in [19], where only the weaker property of the "Representability Axiom (S2)" is generally satisfied; see p.9 of *op. cit.*

As usual, an internal category, K, in $\mathbf{C}$ is given by an object (i.e. a class), $|\mathsf{K}|$, of K-objects, and an internal family, $\{\mathsf{K}(A,B)\}_{A,B:|\mathsf{K}|}$, of K-morphisms indexed by domain and codomain, satisfying the expected axioms for identities and composition, see e.g. [17]. We say that an internal category K in $\mathbf{C}$ is *locally small* if the internal family

$$\{\mathsf{K}(A,B)\}_{A,B:|\mathsf{K}|} \longrightarrow |\mathsf{K}| \times |\mathsf{K}|$$

is a small map in $\mathbf{C}$. It is *small* if, in addition, $|\mathsf{K}|$ is small.

As the definitions above suggest, class structure provides a good framework for addressing size issues in internal categories. For example, we call an internal category K *small-complete* if there is a map $\lim_{\mathsf{K}} : \mathrm{Diagrams}_{\mathsf{K}} \longrightarrow \mathrm{Cones}_{\mathsf{K}}$ in $\mathbf{C}$, where $\mathrm{Diagrams}_{\mathsf{K}}$ is the class of all small diagrams in K and $\mathrm{Cones}_{\mathsf{K}}$ is the class of all small cones, and $\lim_{\mathsf{K}}$ maps each small diagram to a limiting cone for it. N.b. K is *not* required to be small.

An *internal functor*, F, from K to L is given by a morphism

$$F : |\mathsf{K}| \longrightarrow |\mathsf{L}|,$$

expressing the action on objects, together with a family of maps

$$\{F_{A,B} : \mathsf{K}(A,B) \longrightarrow \mathsf{L}(FA,FB)\}_{A,B:|\mathsf{K}|}$$

that preserve identities and composition, again see [17].

We briefly exhibit $\mathbf{S}$ as an internal category in $\mathbf{C}$, before turning attention to $\mathbf{P}$ and $\mathbf{pP}$. The internal category S is defined by

$$|\mathsf{S}| = \mathcal{P}_{\mathbf{S}}U \qquad \mathsf{S}(A,B) = B^A,$$

where the family $\{B^A\}_{A,B:\mathcal{P}_{\mathbf{S}}U}$ is formally defined as an exponential of small objects in the slice category $\mathbf{C}/(\mathcal{P}_{\mathbf{S}}U \times \mathcal{P}_{\mathbf{S}}U)$. Identities and composition are defined in the obvious way. By the remarks on smallness and indexed products

in Section 2, S is a small-complete locally small internal category in $\mathbf{C}$. N.b. it is not a small internal category as neither $|\mathsf{S}|$ nor $\sum_{A,B:|\mathsf{S}|} B^A$ is small.

We next justify the claim that the internal category S indeed internalizes the external category $\mathbf{S}$ within $\mathbf{C}$. This result is a consequence of dependent products $\prod_{i:I} X_i$ existing for small I, together with $\mathcal{P}_{\mathbf{S}}U$ being the carrier of a generic small map in $\mathbf{C}$. Specifically, any map $g\colon I \longrightarrow \mathcal{P}_{\mathbf{S}}U$ in $\mathbf{C}$ is realized as an internal family $f\colon X \longrightarrow I$, with f small, by taking the pullback below, where γ_U is as in (1).

$$\begin{array}{ccc} X & \longrightarrow & \ni_U \\ {\scriptstyle f}\downarrow & \lrcorner & \downarrow{\scriptstyle \gamma_U} \\ I & \xrightarrow[g]{} & \mathcal{P}_{\mathbf{S}}U \end{array} \qquad (2)$$

Conversely, it is shown in [41, Theorem 8] that, for any small map $f\colon X \longrightarrow I$ in $\mathbf{C}$, there exists a canonical (though not unique) $g\colon I \longrightarrow \mathcal{P}_{\mathbf{S}}U$ fitting into a pullback diagram of the form above. In other words, any I-indexed family of small sets in $\mathbf{C}$ determines a corresponding map $I \longrightarrow |\mathsf{S}|$, and vice versa.

The above correpondence is more properly expressed using the theory of fibrations. Define a fibration $\mathbf{Fam}(\mathbf{S}) \to \mathbf{C}$ as follows. The objects of $\mathbf{Fam}(\mathbf{S})$ are the small maps in $\mathbf{C}$. The morphisms between small maps are just those of the arrow category $\mathbf{C}^{\to}$. Then the codomain functor $\mathbf{Fam}(\mathbf{S}) \to \mathbf{C}$ is a fibration (it is the full subfibration of the codomain fibration $\mathbf{C}^{\to} \longrightarrow \mathbf{C}$ on those objects of $\mathbf{C}^{\to}$ that are small maps). Observe that the fibre over $\mathbf{1}$ is isomorphic to $\mathbf{S}$ itself. Recall, from [17, §7.3], the definition of the *externalization* of an internal category K in $\mathbf{C}$ as a split fibration $\mathbf{Ext}(\mathsf{K}) \to \mathbf{C}$.[2] The proposition below is a consequence of the discussion in the previous paragraph.

Proposition 3.1 *The fibration* $\mathbf{Fam}(\mathbf{S}) \to \mathbf{C}$ *is equivalent to the externalization,* $\mathbf{Ext}(\mathsf{S}) \to \mathbf{C}$*, of the internal category* S*.*

We next construe both $\mathbf{P}$ and $\mathbf{pP}$ as internal categories P and pP respectively. First we define P by

$$|\mathsf{P}| = \{A\colon \mathcal{P}_{\mathbf{S}}\, U \mid A \text{ is well-complete}\} \qquad\qquad \mathsf{P}(A,B) = B^A\,.$$

Thus P is an internal full subcategory of S, and hence locally small. The internal category pP is defined by

$$|\mathsf{pP}| \;=\; |\mathsf{P}| \qquad \mathsf{pP}(A,B) \;=\; A \rightharpoonup B\,,$$

[2] Warning: our notation differs from [17], where $\mathbf{Fam}(\mathsf{K})$ is used for the externalization of K. In our case, $\mathbf{Fam}(\mathbf{S}) \to \mathbf{C}$ is not in general a split fibration.

with the obvious identities and composition. Again, pP is locally small (because $A \rightharpoonup B \cong (\mathbb{L}B)^A$).

Again, we demonstrate that these are reasonable definitions, by exhibiting the externalizations of P and pP. Define $\mathbf{Fam}(\mathbf{P})$ to be the full subcategory of $\mathbf{Fam}(\mathbf{S})$ whose objects are those small maps $X \longrightarrow I$ in $\mathbf{C}$ satisfying

$$\mathbf{C} \models \forall i\colon I.\ X_i \text{ is a predomain.}$$

Define $\mathbf{Fam}(\mathbf{pP})$ to have the same objects as $\mathbf{Fam}(\mathbf{P})$, but with a morphism from $X \longrightarrow I$ to $Y \longrightarrow J$ given by a morphism $f\colon I \longrightarrow J$ together with an I-indexed family of partial maps $\{g_i\colon X_i \rightharpoonup Y_{f(i)}\}_{i:I}$. Identities and composition are obvious. The codomain functors $\mathbf{Fam}(\mathbf{P}) \to \mathbf{C}$ and $\mathbf{Fam}(\mathbf{pP}) \to \mathbf{C}$ are both fibrations. The result below is a consequence of Proposition 3.1.

Proposition 3.2

(1) The fibration $\mathbf{Fam}(\mathbf{P}) \to \mathbf{C}$ *is equivalent to* $\mathbf{Ext}(\mathsf{P}) \to \mathbf{C}$.
(2) The fibration $\mathbf{Fam}(\mathbf{pP}) \to \mathbf{C}$ *is equivalent to* $\mathbf{Ext}(\mathsf{pP}) \to \mathbf{C}$.

By the proposition above, fibred structure on $\mathbf{P}$ and $\mathbf{pP}$ lifts to internal structure on the internal categories P and pP. For example, assuming Axiom **1**, the $\mathbb{L}$-monad on $\mathbf{P}$ determines an internal monad $(L, \{\cdot\}, \bigcup)$ on P. Also, Proposition 2.5 can be interpreted as an internal proposition about the internal category pP. Statements (4) and (5) of the proposition together imply that, in the presence of Axiom **1**, it holds that the internal category P is small-complete, with limits inherited from S. On the internal category pP, one can derive internal functors:

$$\mathsf{pP} \times \mathsf{pP} \xrightarrow{\times} \mathsf{pP} \tag{3}$$

$$\mathsf{pP}^{op} \times \mathsf{pP} \xrightarrow{\rightharpoonup} \mathsf{pP} \tag{4}$$

$$\mathsf{pP} \times \mathsf{pP} \xrightarrow{+} \mathsf{pP}\,, \tag{5}$$

where (4) requires Axiom **1**, and (5) requires Axiom **2**. N.b. although $\times$ extends product on P, it is a monoidal rather than cartesian product on pP; whereas $+$ is a genuine binary coproduct functor on pP.

Our goal is to prove the algebraic compactness, in the sense of Freyd [9,10], of the internal category pP. We recall this notion for ordinary categories. Given an endofunctor F on an arbitrary category $\mathcal{K}$, a *bifree* algebra is an initial F-algebra $a\colon FA \longrightarrow A$ for which a^{-1} is also a final F-coalgebra (by Lambek's Lemma, an initial algebra is always an isomorphism). A category $\mathcal{K}$ is said to be *algebraically compact* if every endofunctor on it has a bifree algebra.

The correct formulation of algebraic compactness for an internal category K in $\mathbf{C}$ is slightly subtle because there need not be any object of all K-endofunctors

in $\mathbf{C}$ to allow an internal universal quantification. Instead, we make an external quantification over internal families of internal functors. Technically, this ensures that the definition is stable under the formation of slice categories of $\mathbf{C}$. This property is indispensable as it allows one to derive *parametrized algebraic compactness* in the sense of [3].

Given internal categories K and L in $\mathbf{C}$, and an object I of $\mathbf{C}$, an *I-indexed family of internal functors* from K to L is given by a pair of maps

$$\begin{aligned} I \times |\mathsf{K}| &\longrightarrow |\mathsf{L}| \\ I \times \{\mathsf{K}(A,B)\}_{A,B:|\mathsf{K}|} &\longrightarrow \{\mathsf{L}(A',B')\}_{A',B':|\mathsf{L}|} \end{aligned}$$

satisfying the obvious properties. (Equivalently, an I-indexed family of internal functors is just an internal functor from K to L when lifted to internal categories in the slice category $\mathbf{C}/I$.) We use the convenient notation $\{F_i \colon \mathsf{K} \to \mathsf{L}\}_{i:I}$ for an I-indexed family of functors.

Definition 3.3 (Algebraic compactness) An *algebraically compact* internal category is given by an internal category K together with, for every internal family $\{F_i \colon \mathsf{K} \to \mathsf{K}\}_{i:I}$ in $\mathbf{C}$ of internal endofunctors, an associated morphism $A_{(-)} \colon I \longrightarrow |\mathsf{K}|$ and family $\{a_i : \mathsf{K}(F_i A_i, A_i)\}_{i:I}$ such that:

$$\mathbf{C} \models \forall i : I.\ a_i \text{ is a bifree } F_i\text{-algebra.}$$

Moreover, the above data must be preserved by reindexing: i.e., for $f \colon J \longrightarrow I$ in $\mathbf{C}$, let $B_{(-)} \colon J \longrightarrow |\mathsf{K}|$ and $\{b_j : \mathsf{K}(F_{f(j)} B_j, B_j)\}_{j:J}$ be associated, as above, with the J-indexed family $\{F_{f(j)} \colon \mathsf{K} \to \mathsf{K}\}_{j:J}$; then it must hold that $B_{(-)} = A_{(-)} \circ f$ and $b_{(-)} = a_{(-)} \circ f$.

Proposition 3.4 (Parametrized algebraic compactness) *Suppose that K and L are internal categories with K algebraically compact, and suppose that $F \colon \mathsf{L} \times \mathsf{K} \to \mathsf{K}$ is an internal functor. Viewing F as indexed over $|\mathsf{L}|$, let $A_{(-)} \colon |\mathsf{L}| \longrightarrow |\mathsf{K}|$ and $\{a_B : \mathsf{K}(F(B, A_B), A_B)\}_{B:|\mathsf{L}|}$ be the data given by algebraic compactness. Then there exists a unique internal functor $F^\dagger \colon \mathsf{L} \to \mathsf{K}$ such that $F^\dagger B = A_B$ and $a_B \colon F(B, F^\dagger B) \cong F^\dagger B$ is natural in B.*

Proposition 3.5 *If K, L and L' are internal categories, with K algebraically compact, and $F \colon \mathsf{L}' \times \mathsf{K} \to \mathsf{K}$ and $G \colon \mathsf{L} \to \mathsf{L}'$ are internal functors, then it holds that $(F \circ (G \times Id_{\mathsf{K}}))^\dagger = F^\dagger \circ G : \mathsf{L} \to \mathsf{K}$.*

Proposition 3.4 is self proving. Proposition 3.5, which establishes an *equality* between functors, follows directly from the construction of $F^\dagger$.

We briefly pause to consider the external meaning of the algebraic compactness of an internal category. Recall, from [17, Proposition 7.3.8], that the internal endofunctors on an internal category K are in one-to-one correspondence with

fibred endofunctors on its externalization $\mathbf{Ext}(\mathsf{K}) \to \mathbf{C}$. Thus, if the internal category K is algebraically compact then every fibred endofunctor on the externalization $\mathbf{Ext}(\mathsf{K}) \to \mathbf{C}$ has a fibred bifree algebra (i.e. it has a bifree algebra, and this is preserved by reindexing). However, Definition 3.3 is stronger than this property. It is equivalent to: every fibred endofunctor on every slice of the fibration $\mathbf{Ext}(\mathsf{K}) \to \mathbf{C}$ has a specified bifree algebra; and specified bifree algebras are strictly preserved by reindexing. One can imagine possible weaker definitions with the strictness requirement relaxed. Nonetheless, strictness is useful in practice (for example, to obtain the equality in Proposition 3.5). Moreover, we are indeed able to achieve the full requirements of Definition 3.3 in the first main result of the paper, which we now state.

Theorem 1 *If Axiom* **1** *holds then* pP *is algebraically compact.*

By Proposition 3.2 and the remarks above, this theorem implies the fibred algebraic compactness of every slice fibration of $\mathbf{Fam}(\mathbf{pP}) \to \mathbf{C}$.

The proof of Theorem 1 occupies Sections 4–9. The strategy is to establish a version of the limit-colimit coincidence of classical domain theory (see, e.g. [45]), and apply it to pP. However, a major complication arises. In many models of our setting, the limit-colimit coincidence is false if formulated using diagrams indexed by the natural numbers $\mathbf{N}$, see [27] for a counterexample. We solve this problem by developing a nontrivial variant, under which diagrams are indexed by the carrier $\mathbf{I}$ of the initial-algebra structure for $\mathbb{L}$.

First, in Section 4, we develop a theory of *strict* maps between *pointed* objects, using algebras of the $\mathbb{L}$-monad. Crucial to the proof of Theorem 1 is the identification of a novel notion of (multi)strict dependent family. The major part of Section 4 is devoted to establishing the basic properties of such families.

Next, in Section 5, we develop two basic operations: a "minimum" map $\mathrm{min} : \mathbf{I} \times \mathbf{I} \longrightarrow\!\!\!\!\!\rightarrow \mathbf{I}$, which provides a semilattice structure on $\mathbf{I}$; and a "limit-finding" map $\bigsqcup : X^{\mathbf{I}} \longrightarrow\!\!\!\!\!\rightarrow X$, for any complete object X. With these at our disposal, we formulate and prove our version of the limit-colimit coincidence in Section 6.

In Section 7, we apply the limit-colimit coincidence to show that any *suitable* internal category K is algebraically compact. An important aspect of suitability is that the class of objects $|\mathsf{K}|$ and the hom-classes $\mathsf{K}(A, B)$ must all be pointed. This allows $\mathbf{I}$-indexed diagrams, of the form required by the limit-colimit coincidence, to be constructed, using the initial algebra property of $\mathbf{I}$, and making essential use of the properties of strict dependent families developed in Section 4. The required bifree algebras are then obtained as *bilimits* of the constructed diagrams. Section 8 establishes various well-behavedness properties of this method of constructing bifree algebras.

Finally, in Section 9, we show that pP is indeed a suitable category and hence

algebraically compact. This concludes the proof of Theorem 1.

It is worth contrasting Theorem 1 and Proposition 2.7. Using Freyd's reduction of recursive types to algebraic compactness [9,3], Theorem 1 does guarantee that, for any predomain Y, a solution X to $X \cong X \rightharpoonup Y$ exists in **pP**. Indeed, one can solve arbitrary recursive domain equations in **pP**. In the light of Proposition 2.7, we emphasise the consequences of a universal object that enable this result to be established. Firstly, the universal object allows **pP** be viewed as an internal category. Secondly, and crucially, the initial-algebra property of **I** holds in the category of classes, and hence is applicable to the object $|\mathsf{pP}|$. When carried out in the internal set theory of a category of classes with universal object, the proof that the initial-algebra property holds uses the full IZF axioms of Replacement and Separation. These two principles together are not compatible with the category of sets being an arbitrary elementary topos. Put in more technical terms, it is not possible to embed an arbitrary elementary topos as the full subcategory of small objects in a category of classes with universal object. Indeed, the counterexample used to prove Proposition 2.7 is one elementary topos that has no such embedding.

Although in this paper we use models of IZF set theory to achieve algebraic compactness, many other set theories and type theories appear rich enough to carry out the proofs in this paper. One such theory is the Extended Calculus of Constructions (ECC) [23], as used, for example, in [35]. However, it appears that one does not need the full impredicativity of ECC. In fact, it seems likely that, with appropriate reformulations, the development of this paper could be carried out in the (predicative) context of Martin-Löf's Type Theory [26]. Similarly, it appears that a predicative set theory, in which Ω is a proper class, could be used rather than IZF, for example Aczel's CZF [1]. Indeed the basic axiomatization adapts straightforwardly to a predicative setting provided one assumes to begin with that Σ is a sub*set* of the class Ω. Such possibilities suggest that it is the *conceptual* strength of IZF that we are exploiting in this paper, rather than its *proof-theoretic* strength.

Nevertheless, there are two reasons for being content with the formulation in this paper. Firstly, the categorical description of the models, given by the axioms for class structure [41,43], is very simple. Secondly, and more importantly, the theory, as presented here, is sufficiently general to incorporate an extensive range of domain-theoretic and realizability models, see Section 15.

4 Pointed objects and multistrict maps

As crucial preparation for the proof of Theorem 1, we use the lifting monad to implement a notion of pointed object, and of strict map between pointed

objects. For us, a *pointed object* (X, α) is simply an Eilenberg-Moore algebra for the monad $(\mathbb{L}, \{\cdot\}, \bigcup)$. In other words, $\alpha\colon \mathbb{L}X \longrightarrow X$ must satisfy the *unit* and *multiplication* laws:

$$\alpha(\{x\}) \;=\; x \qquad \text{for all } x\colon X, \tag{6}$$

$$\alpha(\bigcup E) \;=\; \alpha(\{\alpha(e) \mid e \in E\}) \qquad \text{for all } E\colon \mathbb{L}^2X. \tag{7}$$

If $\bot \in \Sigma$ then one can think of $\alpha(\emptyset)$ as the identified "point" of X, but the notion of pointed object also makes sense without the assumption that $\bot \in \Sigma$. A *strict map* $h\colon (X, \alpha) \longrightarrow (Y, \beta)$ between pointed objects is simply an algebra homorphism, i.e. a morphism $h\colon X \longrightarrow Y$ such that $h \circ \alpha = \beta \circ \mathbb{L}h$, which is equivalently stated as:

$$h(\alpha(e)) \;=\; \beta(\{h(x) \mid x \in e\}) \qquad \text{for all } e\colon \mathbb{L}X.$$

Thus the category of pointed objects and strict maps is just the Eilenberg-Moore category for the $\mathbb{L}$ monad on $\mathbf{C}$. In the sequel, we make free use of limits of pointed objects, which are created by the forgetful functor to $\mathbf{C}$. Sometimes we suppress the pointed structure, writing X instead of (X, α), when it is clear from the context.

Equation (7) above equivalently says that α is a strict map from $(\mathbb{L}X, \bigcup)$, which is always pointed, to (X, α). The proposition below shows that strict maps between lifted objects (with pointed structure $\bigcup$) have a natural characterization, which to some extent justifies the "pointed" and "strict" terminologies. In the statement of this proposition, and henceforth, we write $e\!\downarrow$ for the Σ-property $\exists x\colon X.\, x \in e$, for $e\colon \mathbb{L}X$.

Proposition 4.1 *The following are equivalent for any map $f\colon \mathbb{L}X \longrightarrow \mathbb{L}Y$.*

(1) f is strict.
(2) $\mathbf{C} \models \forall e\colon \mathbb{L}X.\, f(e)\!\downarrow \,\rightarrow\, e\!\downarrow$.

PROOF. To show (1) implies (2), suppose f is strict, i.e for all $E\colon \mathbb{L}^2X$, we have $f(\bigcup E) = \bigcup\{f(e) \mid e \in E\}$. Take any $e\colon \mathbb{L}X$, and suppose $f(e)\!\downarrow$, i.e. that $f(e) = \{y\}$ for some $y\colon Y$. We must show that $e\!\downarrow$. Note that $\{e\}$ and $\{e \mid e\!\downarrow\}$ are both elements of $\mathbb{L}^2X$, with $\bigcup\{e\} = \bigcup\{e \mid e\!\downarrow\}$. Thus we have:

$$\begin{aligned}
\{y\} &= \bigcup\{f(e)\} \\
&= f(\bigcup\{e\}) && \text{as } f \text{ strict} \\
&= f(\bigcup\{e \mid e\!\downarrow\}) \\
&= \bigcup\{f(e) \mid e\!\downarrow\} && \text{as } f \text{ strict.}
\end{aligned}$$

So $y \in \{f(e) \mid e\!\downarrow\}$. Thus indeed $e\!\downarrow$.

Conversely, suppose that (2) holds, and take any $E : \mathbb{L}^2 X$. We must show that $f(\bigcup E) = \bigcup\{f(e) \mid e \in E\}$. For the left-to-right inclusion, suppose $y \in f(\bigcup E)$. By (2), there exists $x \in \bigcup E$. Thus $f(\bigcup E) = \{y\}$ and $E = \{\{x\}\}$. So indeed, $\bigcup\{f(e) \mid e \in E\} = f(\{x\}) = f(\bigcup E)$. For the converse inclusion, suppose $y \in \bigcup\{f(e) \mid e \in E\}$. Then $E = \{e\}$ for some e with $f(e) = \{y\}$. By (2), there exists x such that $e = \{x\}$. Thus again, $\bigcup\{f(e) \mid e \in E\} = f(\{x\}) = f(\bigcup E)$. $\square$

Definition 4.2 (k-strict map) Given pointed objects $(X_1, \alpha_1), \ldots, (X_k, \alpha_k)$ and (Y, β), a *k-strict map* is a morphism $h \colon X_1 \times \cdots \times X_k \longrightarrow Y$ such that, for each i with $1 \leq i \leq k$, it holds in $\mathbf{C}$ that

$$\begin{array}{l} \forall x_1 : X_1, \ldots, x_{i-1} : X_{i-1},\ x_{i+1} : X_{i+1}, \ldots, x_k : X_k. \\ \quad x_i \mapsto h(x_1, \ldots, x_k) \text{ is a strict map from } (X_i, \alpha_i) \text{ to } (Y, \beta). \end{array}$$

We use *bistrict* for the case $k = 2$, and *multistrict* if we leave k implicit.

The above definition, exploits the internal logic of $\mathbf{C}$ to formulate k-strictness in the natural way. Nevertheless, we remark that the definition also has a simple category-theoretic formulation using the *(double-)strength* of the $\mathbb{L}$ monad, see, e.g., the definition of *bimorphism* in [16].

Proposition 4.3 *For pointed objects (X_1, α_1), ..., (X_k, α_k) and (Y, β), where $k \geq 1$, any k-strict map $h \colon X_1 \times \cdots \times X_k \longrightarrow Y$ is a strict map from $(X_1, \alpha_1) \times \cdots \times (X_k, \alpha_k)$ to (Y, β).*

This result is a special case of Proposition 4.8, which is proved below. In fact, the above proposition holds, more generally, for all *relevant monads* in the sense of [16].

The initial algebra $\mathbf{I}$ of the endofunctor $\mathbb{L}$ on $\mathbf{C}$ carries a pointed structure $\phi = \sigma \circ \bigcup \circ \mathbb{L}\sigma^{-1} \colon \mathbb{L}\mathbf{I} \longrightarrow \mathbf{I}$. The pointed structure on $\mathbf{I}$ interacts nicely with the initial algebra property. Define a "successor" function $s = \sigma \circ \{\cdot\} \colon \mathbf{I} \longrightarrow \mathbf{I}$.

Proposition 4.4 *Suppose that (X, α) is a pointed object and that $f : X \longrightarrow X$ is any (not necessarily strict) morphism. Then, for every $k \geq 1$, there exists a unique k-strict map $h : \mathbf{I}^k \longrightarrow X$ such that the diagram below commutes.*

$$\begin{array}{ccc} \overbrace{\mathbf{I} \times \cdots \times \mathbf{I}}^{k} & \xrightarrow{h} & X \\ {\scriptstyle s \times \cdots \times s}\downarrow & & \downarrow{\scriptstyle f} \\ \mathbf{I} \times \cdots \times \mathbf{I} & \xrightarrow{h} & X \end{array}$$

The case $k = 1$ is proved as [19, Theorem A.5], and we henceforth assume this case. We do not prove the generalization to $k \geq 2$ at this point, as it follows from Proposition 4.9 below.

We now embark on the main technical contribution of this section, the generalization of the notions of strictness and multistrictness to dependent families. The definitions and results are fundamental to our proof of Theorem 1. However, the development may also be of independent interest. It exposes a subtle interplay between dependent types and algebras for the $\mathbb{L}$ monad, which appears to depend heavily on properties of $\mathbb{L}$ that are peculiar to lifting monads. It might be an interesting mathematical project to obtain a more abstract account of the theory that follows.

Definition 4.5 (Strict family) Given an internal family $\{(Y_x, \beta_x)\}_{x:X}$ of pointed objects, where (X, α) is pointed, we say that $\{y_x : Y_x\}_{x:X}$ is a *strict family* if, for all $e : \mathbb{L}X$,

$$y_{\alpha(e)} \;=\; \beta_{\alpha(e)}(\{y_x \mid x \in e\}). \tag{8}$$

To see that this definition makes sense, we must show that $\{y_x \mid x \in e\}$ is a Σ-subterminal subset of $\mathbb{L}(Y_{\alpha(e)})$. It is Σ-subterminal because e is. Moreover, given $x \in e$, we have $e = \{x\}$, whence $\alpha(e) = x$ by (6). Thus indeed $y_x \in Y_{\alpha(e)}$.

Observe that, for a constant family $\{(Y, \beta)\}_{x:X}$, a strict family $\{y_x : Y\}_{x:X}$ is just a strict map $x \mapsto y_x$ from X to Y.

Strict families compose in the natural way.

Lemma 4.6 *Given families $\{(Y_x, \beta_x)\}_{x:X}$ and $\{(Z_{x\,y}, \gamma_{x\,y})\}_{x:X,\, y:Y_x}$ of pointed objects, where (X, α) is pointed, suppose that $\{y_x : Y_x\}_{x:X}$ is a strict family and $\{z_{x\,y} : Z_{x\,y}\}_{y \in Y_x}$ is a strict family for every $x : X$. Then $\{z_{x\,y_x} : Z_{x\,y_x}\}_{x:X}$ is a strict family.*

PROOF. Suppose $\{y_x : Y_x\}_{x:X}$ is a strict family, and $\{z_{x\,y} : Z_{x\,y}\}_{y:Y_x}$ is a strict family for all $x : X$. For any $e : \mathbb{L}X$, define $b_e = \beta_{\alpha(e)}(\{y_x \mid x \in e\})$. Then:

$$\begin{aligned}
z_{\alpha(e)\,y_{\alpha(e)}} &= z_{\alpha(e)\,b_e} && \text{as } y_{(-)} \text{ is strict} \\
&= \gamma_{\alpha(e)\,b_e}(\{z_{\alpha(e)\,y} \mid y \in \{y_x \mid x \in e\}\}) && \text{as } z_{\alpha(e)\,(-)} \text{ is strict} \\
&= \gamma_{\alpha(e)\,y_{\alpha(e)}}(\{z_{\alpha(e)\,y_x} \mid x \in e\}) && \text{as } y_{(-)} \text{ is strict} \\
&= \gamma_{\alpha(e)\,y_{\alpha(e)}}(\{z_{x\,y_x} \mid x \in e\}) && \text{as } x \in e \text{ implies } x = \alpha(e).
\end{aligned}$$

□

Definition 4.7 (k-strict family) Given a family of pointed objects,

$$\{(Y_{x_1\ldots x_k},\, \beta_{x_1\ldots x_k})\}_{x_1 : X_1, \ldots, x_k : X_k}\ ,$$

where $(X_1, \alpha_1), \ldots, (X_k, \alpha_k)$ are pointed, then $\{y_{x_1\ldots x_k} : Y_{x_1\ldots x_k}\}_{x_1 : X_1, \ldots, x_k : X_k}$ is a *k-strict family* if, for each i with $1 \leq i \leq k$, it holds in $\mathbf{C}$ that

$$\forall x_1, \ldots, x_{i-1},\, x_{i+1}, \ldots, x_k.\ \ \{y_{x_1\ldots x_k} : Y_{x_1\ldots x_k}\}_{x_i : X_i} \text{ is a strict family.}$$

Again, for a constant family $\{(Y, \beta)\}_{x_1, \ldots, x_k}$, a k-strict family $\{y_{x_1\ldots x_k} : Y\}_{x_1, \ldots, x_k}$ is just a k-strict map $(x_1, \ldots, x_k) \mapsto y_{x_1\ldots x_k}$ from $X_1 \times \cdots \times X_k$ to Y.

The two propositions below, which are the main results of this section, generalize Propositions 4.3 and 4.4 above to apply to k-strict families.

Proposition 4.8 *If $\{(Y_{x_1\ldots x_k},\, \beta_{x_1\ldots x_k})\}_{x_1 : X_1, \ldots, x_k : X_k}$ and $(X_1, \alpha_1), \ldots, (X_k, \alpha_k)$ are as in Definition 4.7, and $\{y_{x_1\ldots x_k} : Y_{x_1\ldots x_k}\}_{x_1 : X_1, \ldots, x_k : X_k}$ is a k-strict family then $\{y_{x_1\ldots x_k} : Y_{x_1\ldots x_k}\}_{(x_1, \ldots, x_k) : X_1 \times \cdots \times X_k}$ is a strict family.*

By instantiating $\{(Y_{x_1\ldots x_k},\, \beta_{x_1\ldots x_k})\}_{x_1 : X_1, \ldots, x_k : X_k}$ as a constant family, Proposition 4.3 is easily seen to follow as a special case.

PROOF. We prove the case $k = 2$, from which the full result easily follows. Suppose then that $\{y(x_1, x_2) : Y_{x_1 x_2}\}_{x_1, x_2}$ is bistrict. The algebra structure on $X_1 \times X_2$ is $\gamma : \mathbb{L}(X_1 \times X_2) \longrightarrow X_1 \times X_2$ defined by $\gamma(e) = (a_1, a_2)$ where

$$\begin{aligned} a_1 &= \alpha_1(\{x_1 \mid (x_1, x_2) \in e\}) \\ a_2 &= \alpha_2(\{x_2 \mid (x_1, x_2) \in e\}). \end{aligned}$$

We must show that $y(a_1, a_2) = \beta_{a_1 a_2}(\{y(x_1, x_2) \mid (x_1, x_2) \in e\})$. But

$$\begin{aligned} y(a_1, a_2) &= y(\alpha_1(\{x_1 \mid (x_1, x_2) \in e\}), a_2) && \text{def. of } a_1 \\ &= \beta_{a_1 a_2}(\{y(x_1, a_2) \mid (x_1, x_2) \in e\}) && \text{bistrictness} \\ &= \beta_{a_1 a_2}(\{y(x_1, \alpha_2(\{x_2' \mid (x_1', x_2') \in e\})) \mid (x_1, x_2) \in e\}) && \text{def. of } a_2 \\ &= \beta_{a_1 a_2}(\{\beta_{x_1 a_2}(\{y(x_1, x_2') \mid (x_1', x_2') \in e\}) \mid (x_1, x_2) \in e\}) && \text{bistrictness.} \end{aligned}$$

To see that this is equal to $\beta_{a_1 a_2}(\{y(x_1, x_2) \mid (x_1, x_2) \in e\})$, it suffices to verify

$$\{\beta_{x_1 a_2}(\{y(x_1, x_2') \mid (x_1', x_2') \in e\}) \mid (x_1, x_2) \in e\} \ =\ \{y(x_1, x_2) \mid (x_1, x_2) \in e\}.$$

If $(x_1, x_2) \in e$ then $e = \{(x_1, x_2)\}$, and $a_2 = x_2$ by (6). Thus the r.h.s. above is equal to $\{y(x_1, x_2)\}$ and the l.h.s. is equal to $\{\beta_{x_1 x_2}(\{y(x_1, x_2)\})\} = \{y(x_1, x_2)\}$, by (6). Thus the equation holds if $(x_1, x_2) \in e$. It follows that each side is a subset of the other. Hence the equation holds. $\square$

Proposition 4.9 *For internal families*

$$\{(Y_{i_1\ldots i_k},\, \beta_{i_1\ldots i_k})\}_{i_1:\mathbf{I},\ldots,i_k:\mathbf{I}},$$
$$\{f_{i_1\ldots i_k}: Y_{i_1\ldots i_k} \to Y_{si_1\ldots si_k}\}_{i_1:\mathbf{I},\ldots,i_k:\mathbf{I}},$$

of pointed objects and arbitrary functions respectively, there exists a unique k-strict family $y_{(-)\ldots(-)}\colon \prod_{i_1:\mathbf{I}}\cdots\prod_{i_k:\mathbf{I}} Y_{i_1\ldots i_k}$ *satisfying*

$$y_{si_1\ldots si_k} \;=\; f_{i_1\ldots i_k}(y_{i_1\ldots i_k})\,. \tag{9}$$

Again, by taking $\{(Y_{i_1\ldots i_k},\, \beta_{i_1\ldots i_k})\}_{i_1,\ldots,i_k}$ and $\{f_{i_1\ldots i_k}: Y_{i_1\ldots i_k} \to Y_{si_1\ldots si_k}\}_{i_1,\ldots,i_k}$ as constant families, Proposition 4.4 follows.

The remainder of the section is devoted to the proof of Proposition 4.9.

Lemma 4.10 *Given an internal family* $\{(Y_x, \beta_x)\}_{x:X}$ *of pointed objects, with* (X, α) *pointed, then* $(\sum_{x:X} Y_x,\, \gamma)$ *is pointed, where* $\gamma\colon \mathbb{L}(\sum_{x:X} Y_x) \longrightarrow \sum_{x:X} Y_x$ *is defined by* $\gamma(e) = (\gamma_1(e), \gamma_2(e))$*, where*

$$\begin{aligned}\gamma_1(e) \;&=\; \alpha\{x \mid (x,y) \in e\}\\ \gamma_2(e) \;&=\; \beta_{\gamma_1(e)}\{y \mid (\gamma_1(e), y) \in e\}.\end{aligned}$$

Moreover the projection $\pi_1\colon (\sum_{x:X} Y_x,\, \gamma) \longrightarrow (X, \alpha)$ *is strict.*

PROOF. To see that γ is well-defined, one establishes: first, that $\gamma_1(e)$ is well-defined, because $\{x \mid (x,y) \in e\}$ is a Σ-subterminal subset of X; and, second, that $\gamma_2(e)$ is well-defined, because $\{y \mid (\gamma_1(e), y) \in e\}$ is a Σ-subterminal subset of $Y_{\gamma_1(e)}$. In each case, the Σ-subterminal property follows from the same property of e. Note also that the strictness of the projection π_1 is immediate from the definition of $\gamma_1(e)$. Thus it remains only to show that γ satisfies the unit law, (6), and multiplication law, (7).

For the unit law, $\gamma_1(\{(x,y)\}) = x$, by (6) for α; whence $\gamma_2(\{(x,y)\}) = y$, by (6) for β_x. Thus indeed $\gamma(\{(x,y)\}) = (x,y)$.

For the multiplication law, take any $E\colon \mathbb{L}^2(\sum_{x:X} Y_x)$. We must show that $\gamma(\bigcup E) = \gamma(\{\gamma(e) \mid e \in E\})$. For the first components, define $E_1\colon \mathbb{L}^2 X$ by

$$E_1 \;=\; \{\{x \mid (x,y) \in e\} \mid e \in E\}.$$

Then we have:

$$\begin{aligned}\gamma_1(\{\gamma(e) \mid e \in E\}) \;&=\; \alpha(\{x \mid (x,y) \in \{\gamma(e) \mid e \in E\}\}) && \text{def. of } \gamma_1\\ &=\; \alpha(\{\gamma_1(e) \mid e \in E\}) && \text{def. of } \gamma\\ &=\; \alpha(\{\alpha(\{x \mid (x,y) \in e\}) \mid e \in E\} && \text{def. of } \gamma_1\end{aligned}$$

$$
\begin{aligned}
&= \alpha(\{\alpha(e_1) \mid e_1 \in E_1\}) && \text{def. of } E_1\\
&= \alpha(\textstyle\bigcup E_1) && \text{by (7) for } \alpha\\
&= \alpha\{x \mid (x,y) \in \textstyle\bigcup E\} && \text{def. of } E_1\\
&= \gamma_1(\textstyle\bigcup E) && \text{def. of } \gamma_1.
\end{aligned}
$$

So the first components agree. Write a for this value, and observe that

$$e \in E \text{ implies } \gamma_1(e) = a, \tag{10}$$
$$(x,y) \in \textstyle\bigcup E \text{ implies } x = a, \tag{11}$$

where the latter property is by (6) for α.

To show that the second components agree, define $E_2 : \mathbb{L}^2(Y_a)$ by

$$E_2 \;=\; \{\{y \mid (a,y) \in e\} \mid e \in E\},$$

where the Σ-subterminal property of $\{y \mid (a,y) \in e\}$ holds by (11). Then:

$$
\begin{aligned}
\gamma_2(\{\gamma(e) \mid e \in E\}) &= \beta_a(\{y \mid (a,y) \in \{\gamma(e) \mid e \in E\}\}) && \text{def. of } \gamma_2\\
&= \beta_a(\{\gamma_2(e) \mid e \in E\}) && \text{by (10)}\\
&= \beta_a(\{\beta_a(\{y \mid (a,y) \in e)\}) \mid e \in E) && \text{def. of } \gamma_2\\
&= \beta_a(\{\beta_a(e_2) \mid e_2 \in E_2\}) && \text{def. of } E_2\\
&= \beta_a(\textstyle\bigcup E_2) && \text{by (7) for } \beta_a\\
&= \beta_a(\{y \mid (a,y) \in \textstyle\bigcup E\}) && \text{def. of } E_2\\
&= \gamma_2(\textstyle\bigcup E) && \text{def. of } \gamma_2.
\end{aligned}
$$

□

Lemma 4.11 *Given $\{(Y_x, \beta_x)\}_{x:X}$ and (X,α) as in Definition 4.5, the following are equivalent for any $\{y_x : Y_x\}_{x:X}$.*

(1) $y_{(-)}$ is a strict family.
(2) The map $x \mapsto (x, y_x) : (X,\alpha) \longrightarrow (\sum_{x:X} Y_x,\, \gamma)$ is strict, where γ is as in Lemma 4.10.

PROOF. For any $e : \mathbb{L}X$, we have, by the definition of γ in Lemma 4.10, that

$$\gamma_1(\{(x, y_x) \mid x \in e\}) \;=\; \alpha(e)$$

and

$$
\begin{aligned}
\gamma_2(\{(x,y_x) \mid x \in e\}) &= \beta_{\alpha(e)}(\{y_{\alpha(e)} \mid \alpha(e) \in e\}) && \text{def. of } \gamma_2\\
&= \beta_{\alpha(e)}(\{y_x \mid x \in e\}) && \text{as } x \in e \text{ implies } x = \alpha(e).
\end{aligned}
$$

It follows immediately that $y_{\alpha(e)} = \beta_{\alpha(e)}(\{y_x \mid x \in e\})$ if and only if $(\alpha(e), y_{\alpha_e}) = \gamma(\{(x, y_x) \mid x \in e\})$; i.e. statements (1) and (2) are equivalent. $\Box$

We remark that Lemma 4.11 does not extend to give a characterisation of k-strict families in terms of ordinary k-strictness. Indeed, given a k-strict family $\{y_{x_1\ldots x_k} \colon Y_{x_1\ldots x_k}\}_{x_1 \colon X_1, \ldots, x_k \colon X_k}$ it is not necessarily the case that the map

$$(x_1, \ldots, x_k) \mapsto (x_1, \ldots, x_k, y) \colon X_1 \times \cdots \times X_k \longrightarrow \sum_{(x_1,\ldots,x_k) \colon X_1 \times \cdots \times X_k} Y_{x_1\ldots x_k}$$

is k-strict. Intuitively, one needs to replace $X_1 \times \cdots \times X_k$ here with a tensor product $X_1 \otimes \cdots \otimes X_k$. But such an approach leads to indexing problems, because $Y_{x_1\ldots x_k}$ is indexed over the product $X_1 \times \cdots \times X_k$. Instead, Lemma 4.13 below provides a correct characterization of multistrict families.

Lemma 4.12 *Given $\{(Y_x, \beta_x)\}_{x \colon X}$ and (X, α) as in Definition 4.5 with X small, define*

$$\prod_{x \colon X}^{\circ} Y_x \;=\; \{y_{(-)} \colon \prod_{x \colon X} Y_x \mid y_{(-)} \text{ is a strict family}\}.$$

Then $(\prod^{\circ}_{x \colon X} Y_x \,, \pi^{\circ})$ is pointed, with $\pi^{\circ} \colon \mathbb{L}(\prod^{\circ}_{x \colon X} Y_x) \longrightarrow \prod^{\circ}_{x \colon X} Y_x$ defined by:

$$(\pi^{\circ}(e))_x \;=\; \beta_x(\{y_x \mid y_{(-)} \in e\}) \qquad \text{for } e \colon \prod_{x \colon X}^{\circ} Y_x \text{ and } x \colon X.$$

PROOF. First we show that $(\pi^{\circ}(e))_{(-)}$ is indeed a strict family, i.e. that $(\pi^{\circ}(e))_{\alpha(e_1)} = \beta_{\alpha(e_1)}(\{(\pi^{\circ}(e))_{(x)} \mid x \in e_1\})$, for $e_1 \colon \mathbb{L}X$. Because $x \in e_1$ implies $x = \alpha(e_1)$, we have that

$$E' \;=\; \{\{y_x \mid x \in e_1\} \mid y_{(-)} \in e\} \qquad E'' \;=\; \{\{y_x \mid y_{(-)} \in e\} \mid x \in e_1\}$$

are both in $\mathbb{L}^2(Y_{\alpha(e_1)})$. Moreover $\bigcup E' = \bigcup E''$. Then:

$$\begin{array}{lll}
(\pi^{\circ}(e))_{\alpha(e_1)} &= \beta_{\alpha(e_1)}(\{y_{\alpha(e_1)} \mid y_{(-)} \in e\}) & \text{def. of } \pi^{\circ}(e) \\
&= \beta_{\alpha(e_1)}(\{\beta_{\alpha(e_1)}(\{y_x \mid x \in e_1\}) \mid y_{(-)} \in e\}) & \text{as } y_{(-)} \text{ strict} \\
&= \beta_{\alpha(e_1)}(\{\beta_{\alpha(e_1)}(e') \mid e' \in E'\}) & \text{def. of } E' \\
&= \beta_{\alpha(e_1)}(\bigcup E') & \text{by (7) for } \beta_{\alpha(e_1)} \\
&= \beta_{\alpha(e_1)}(\bigcup E'') & \text{as } \bigcup E' = \bigcup E'' \\
&= \beta_{\alpha(e_1)}(\{\beta_{\alpha(e_1)}(e'') \mid e'' \in E''\}) & \text{by (7) for } \beta_{\alpha(e_1)} \\
&= \beta_{\alpha(e_1)}(\{\beta_{\alpha(e_1)}(\{y_x \mid y_{(-)} \in e\}) \mid x \in e_1\}) & \text{def. of } E'' \\
&= \beta_{\alpha(e_1)}(\{\beta_x(\{y_x \mid y_{(-)} \in e\}) \mid x \in e_1\}) & \text{as } x \in e_1 \text{ implies } x = \alpha(e_1) \\
&= \beta_{\alpha(e_1)}(\{(\pi^{\circ}(e))_{(x)} \mid x \in e_1\}) & \text{def. of } \pi^{\circ}(e).
\end{array}$$

For the unit law, (6), one must show $(\pi^\circ(\{y_{(-)}\}))_x = y_x$, which is easy.

For the multiplication law, (7), we must show that $(\pi^\circ(\{\pi^\circ(e) \mid e \in E\}))_x = (\pi^\circ(\bigcup E))_x$ for $x : X$ and $E : \mathbb{L}^2(\prod^\circ_{x:Y} Y_x)$. Define $E_x : \mathbb{L}^2(Y_x)$ by

$$E_x \;=\; \{\{y_x \mid y_{(-)} \in e\} \mid e \in E\}.$$

Then indeed,

$$\begin{array}{lcll}
(\pi^\circ(\{\pi^\circ(e) \mid e \in E\}))_x & = & \beta_x(\{(\pi^\circ(e))_x \mid e \in E\})\}) & \text{def. of } \pi^\circ \\
& = & \beta_x(\{\beta_x(\{y_x \mid y_{(-)} \in e\}) \mid e \in E\})\}) & \text{def. of } \pi^\circ \\
& = & \beta_x(\{\beta_x(e_x) \mid e_x \in E_x\})\}) & \text{def. of } E_x \\
& = & \beta_x(\bigcup E_x) & \text{by (7) for } \beta_x \\
& = & \beta_x(\{y_x \mid y_{(-)} \in \bigcup E\}) & \text{def. of } E_x \\
& = & (\pi^\circ(\bigcup E))_x & \text{def. of } \pi^\circ.
\end{array}$$

□

Henceforth (e.g. in statements 2 and 3 of the lemma below), the algebra structure of an object of the form $\prod^\circ_{x:X} Y_x$ is always taken to be as above. Also, when $\{(Y, \beta)\}_{x:X}$ is a constant family, we write $X \multimap Y$ for $\prod^\circ_{x:X} Y$. Thus Lemma 4.12 shows that the object $X \multimap Y$, of all strict functions from X to Y, is pointed.

Lemma 4.13 *Given* $\{(Y_{x_1 \ldots x_k},\, \beta_{x_1 \ldots x_k})\}_{x_1 : X_1, \ldots, x_k : X_k}$ *and* $(X_1, \alpha_1), \ldots, (X_k, \alpha_k)$ *as in Definition 4.7 with* $X_1, \ldots, X_k$ *small, then, for* $k \geq 1$*, the following are equivalent.*

(1) $y_{(-)\ldots(-)} : \prod_{x_1 : X_1} \cdots \prod_{x_k : X_k} Y_{x_1 \ldots x_k}$*, is a* k*-strict family.*
(2) $\{y_{x_1 \ldots x_{k-1}(-)}\}_{x_1 : X_1, \ldots, x_{k-1} : X_{k-1}} : \prod_{x_1 : X_1} \cdots \prod_{x_{k-1} : X_{k-1}} (\prod^\circ_{x_k : X_k} Y_{x_1 \ldots x_k})$ *is a* $(k-1)$*-strict family.*
(3) $y_{(-)\ldots(-)} \in \prod^\circ_{x_1 : X_1} \cdots \prod^\circ_{x_k : X_k} Y_{x_1 \ldots x_k}$.

PROOF. Statement (1) holds if and only if, for each i with $1 \leq i \leq k$,

$$y_{\mathbf{x}\,\alpha(e_i)\,\mathbf{x}'} \;=\; \beta_{\mathbf{x}\,\alpha(e_i)\,\mathbf{x}'}(\{y_{\mathbf{x}\,x_i\,\mathbf{x}'} \mid x_i \in e_i\})\,, \tag{12}$$

where $e_i : \mathbb{L}(X_i)$, and $\mathbf{x}$ and $\mathbf{x}'$ are vectors of elements $x_1 \ldots x_{i-1}$ and $x_{i+1} \ldots x_k$ respectively, with each $x_j : X_j$.

Similarly, statement (2) holds if and only if:

$$y_{\mathbf{x}\,\alpha(e_k)} \;=\; \beta_{\mathbf{x}\,\alpha(e_k)}(\{y_{\mathbf{x}\,x_k} \mid x_k \in e_k\})\,, \tag{13}$$

i.e. each $y_{\mathbf{x}\,(-)}$ is a strict family; and also, for each i with $1 \leq i \leq k-1$,

$$y_{\mathbf{x}\,\alpha(e_i)\,\mathbf{x}''\,(-)} \;=\; \pi^{\circ}_{\mathbf{x}\,\alpha(e_i)\,\mathbf{x}''}(\{y_{\mathbf{x}\,x_i\,\mathbf{x}''\,(-)} \mid x_i \in e_i\})\,, \tag{14}$$

where $\mathbf{x}''$ is a vector of elements $x_{i+1} \ldots x_{k-1}$, and $\pi^{\circ}_{\mathbf{x}\,\alpha(e_i)\,\mathbf{x}''}$ is the algebra on $\prod^{\circ}_{x_k : X_k} Y_{\mathbf{x}\,\alpha(e_i)\,\mathbf{x}''\,x_k}$.

However, equation (13) is exactly the $i = k$ case of equation (12). Further,

$$(\pi^{\circ}_{\mathbf{x}\,\alpha(e_i)\,\mathbf{x}''}(\{y_{\mathbf{x}\,x_i\,\mathbf{x}''\,(-)} \mid x_i \in e_i\}))_{x_k} \;=\; \beta_{\mathbf{x}\,\alpha(e_i)\,\mathbf{x}''\,x_k}(\{y_{\mathbf{x}\,x_i\,\mathbf{x}''\,x_k} \mid x_i \in e_i\}),$$

by the definition of π°. So, for $i < k$, (14) holds if and only if (12) holds. Therefore statement (1) is indeed equivalent to statement (2).

The equivalence of statements (1) and (3) is proved by a straightforward k-fold iterated application of the equivalence between statements (1) and (2). □

Lemma 4.14 *Suppose that (X, α) is pointed, $f, g\colon \mathbf{I} \longrightarrow X$ are strict and $f \circ s = g \circ s$ then $f = g$.*

PROOF. One easily shows that the assumptions imply that $\sigma\colon \mathbb{L}\mathbf{I} \longrightarrow \mathbf{I}$ restricts to a subalgebra $\sigma'\colon \mathbb{L}\mathbf{I}' \longrightarrow \mathbf{I}'$ where $\mathbf{I}' = \{i : \mathbf{I} \mid f(i) = g(i)\}$. Then, because σ is the initial $\mathbb{L}$-algebra, $\mathbf{I}' = \mathbf{I}$. □

PROOF of Proposition 4.9. By induction on k.

When $k = 1$, consider the map

$$g \;=\; \sum_{j:\mathbf{I}} Y_j \xrightarrow{(j,z) \mapsto (sj,\, f_j(z))} \sum_{j:\mathbf{I}} Y_j\,.$$

We establish that strict families $y_{(-)}$ satisfying $y_{si} = f_i(y_i)$ are in one-to-one correspondence with strict $h\colon \mathbf{I} \longrightarrow \sum_{j:\mathbf{I}} Y_j$ satisfying $g \circ h = h \circ s$. Thus, by the $k = 1$ case of Proposition 4.4, there is indeed a unique such family.

Given any strict $h\colon \mathbf{I} \longrightarrow \sum_{j:\mathbf{I}} Y_j$ satisfying $g \circ h = h \circ s$, the composite $\pi_1 \circ h\colon \mathbf{I} \longrightarrow \mathbf{I}$ is strict, by Lemma 4.10, and also $\pi_1 \circ h \circ s = \pi_1 \circ g \circ h = s \circ \pi_1 \circ h$, by the definition of g. So, by the $k = 1$ case of Proposition 4.4, $\pi_1 \circ h = \mathsf{id}_{\mathbf{I}}$. Hence, any such h is of the form $j \mapsto (j, y_j)$ where, by Lemma 4.11, $y_{(-)}$ is a strict family. Moreover, the equation $y_{si} = f_i(y_i)$ follows from $g \circ h = h \circ s$, by the definintion of g. Conversely, given a strict family $y_{(-)}$ satisfying $y_{si} = f_i(y_i)$, the map $h\colon j \mapsto (j, y_j)$ is strict, by Lemma 4.11, and satisfies $g \circ h = h \circ s$.

When $k > 1$, applying the induction hypothesis, let

$$\{y_{i_1 \ldots i_{k-1}\,(-)}\}_{i_1,\ldots,i_{k-1}}\colon \prod_{i_1:\mathbf{I}} \cdots \prod_{i_{k-1}:\mathbf{I}} (\prod^{\circ}_{i_k:\mathbf{I}} Y_{i_1 \ldots i_k})$$

be the unique $(k-1)$-strict family satisfying

$$(j \mapsto y_{si_1 \dots si_{k-1} j}) \;=\; g_{i_1 \dots i_{k-1}}(j \mapsto y_{i_1 \dots i_{k-1} j})\,, \tag{15}$$

where the family

$$\left\{ g_{i_1 \dots i_{k-1}} \colon (\prod_{j:\mathbf{I}}^{\circ} Y_{i_1 \dots i_{k-1} j}) \to (\prod_{j:\mathbf{I}}^{\circ} Y_{si_1 \dots si_{k-1} j}) \right\}_{i_1, \dots, i_{k-1}},$$

is defined as follows.

Given $z_{(-)} \colon \prod_{j:\mathbf{I}}^{\circ} Y_{i_1 \dots i_{k-1} j}$, consider the composite below,

$$h \;=\; \mathbf{I} \xrightarrow{\sigma^{-1}} \mathbb{L}\mathbf{I} \xrightarrow{\mathbb{L}(j \mapsto (sj,\, f_{i_1 \dots i_{k-1} j}(z_j)))} \mathbb{L}(\sum_{j:\mathbf{I}} Y_{si_1 \dots si_{k-1} j}) \xrightarrow{\gamma} \sum_{j:\mathbf{I}} Y_{si_1 \dots si_{k-1} j}\,,$$

where γ is the pointed structure on $\sum_{j:\mathbf{I}} Y_{si_1 \dots si_{k-1} j}$ from Lemma 4.10. Each component in the above composite is strict (the first is an isomorphism, the second is a lifted map, the third is an algebra structure map). Hence h is itself strict. Consider $\pi_1 \circ h \colon \mathbf{I} \longrightarrow \mathbf{I}$. We have:

$$\begin{aligned}
\pi_1 \circ h \;&=\; \pi_1 \circ \gamma \circ \mathbb{L}(j \mapsto (sj,\, f_{i_1 \dots i_k j}(z_j))) \circ \sigma^{-1} \\
&=\; \phi \circ \mathbb{L}(\pi_1) \circ \mathbb{L}(j \mapsto (sj,\, f_{i_1 \dots i_k j}(z_j))) \circ \sigma^{-1} && \text{as } \pi_1 \text{ is strict} \\
&=\; \phi \circ \mathbb{L}(s) \circ \sigma^{-1} \\
&=\; (\sigma \circ \bigcup \circ\, \mathbb{L}\sigma^{-1}) \circ \mathbb{L}(\sigma \circ \{\cdot\}) \circ \sigma^{-1} && \text{defns. of } \phi \text{ and } s \\
&=\; \sigma \circ (\bigcup \circ \mathbb{L}\{\cdot\}) \circ \sigma^{-1} \;=\; \mathsf{id}_{\mathbf{I}} && \text{as } \bigcup \circ \mathbb{L}\{\cdot\} = \mathsf{id}_{\mathbb{L}\mathbf{I}}.
\end{aligned}$$

Thus, for any $j \colon \mathbf{I}$, we have $h(j) = (j,\, w_{z_{(-)}\, j})$ where $w_{z_{(-)}\, j} \colon Y_{si_1 \dots si_{k-1} j}$. As h is strict, we have by Lemma 4.11, that $(j \mapsto w_{z_{(-)}\, j}) \colon \prod_{j:\mathbf{I}} Y_{si_1 \dots si_{k-1} j}$ is a strict family. So define $g_{i_1 \dots i_{k-1}}(z_{(-)}) \;=\; (j \mapsto w_{z_{(-)}\, j})$.

Observe that

$$\begin{aligned}
h \circ s \;&=\; \gamma \circ \mathbb{L}(j \mapsto (sj,\, f_{i_1 \dots i_{k-1} j}(z_j))) \circ \sigma^{-1} \circ \sigma \circ \{\cdot\} \\
&=\; \gamma \circ \mathbb{L}(j \mapsto (sj,\, f_{i_1 \dots i_{k-1} j}(z_j))) \circ \{\cdot\} \\
&=\; \gamma \circ \{\cdot\} \circ (j \mapsto (sj,\, f_{i_1 \dots i_{k-1} j}(z_j))) \\
&=\; (j \mapsto (sj,\, f_{i_1 \dots i_{k-1} j}(z_j))).
\end{aligned}$$

Thus

$$(g_{i_1 \dots i_{k-1}}(z_{(-)}))(sj) \;=\; f_{i_1 \dots i_{k-1} j}(z_j). \tag{16}$$

We must show that $y_{(-)\dots(-)}$ is the unique k-strict family satisfying (9). As $\{y_{i_1 \dots i_{k-1}(-)}\}_{i_1, \dots, i_{k-1}} \colon \prod_{i_1 : \mathbf{I}} \cdots \prod_{i_{k-1} : \mathbf{I}} (\prod_{i_k : \mathbf{I}}^{\circ} Y_{i_1 \dots i_k})$ is $(k-1)$-strict, we have, by Lemma 4.13, that $y_{(-)\dots(-)} \colon \prod_{i_1 : \mathbf{I}} \cdots \prod_{i_k : \mathbf{I}} Y_{i_1 \dots i_k}$ is k-strict. To see that (9)

holds, we have

$$\begin{aligned} y_{si_1 \dots si_k} &= (g_{i_1 \dots i_k}(y_{i_1 \dots i_{k-1}(-)}))(si_k) && \text{by (15)} \\ &= f_{i_1 \dots i_k}(y_{i_1 \dots i_k}) && \text{by (16).} \end{aligned}$$

It remains to show that $y_{(-)\dots(-)}$ is unique. Suppose that $y'_{(-)\dots(-)}$ is any k-strict family satisfying (9′) (where we write (9′) to mean (9) with y replaced by y'). By Lemma 4.13, $\{y'_{i_1 \dots i_{k-1}(-)}\}_{i_1,\dots,i_{k-1}} : \prod_{i_1 : \mathbf{I}} \cdots \prod_{i_{k-1} : \mathbf{I}} (\prod^{\circ}_{i_k : \mathbf{I}} Y_{i_1 \dots i_k})$ is a $(k-1)$-strict family. It suffices to show that $\{y'_{i_1 \dots i_{k-1}(-)}\}_{i_1,\dots,i_{k-1}}$ satisfies (15′). As both $j \mapsto y'_{si_1 \dots si_{k-1} j}$ and $g_{i_1 \dots i_{k-1}}(j \mapsto y'_{i_1 \dots i_{k-1} j})$ are strict, it suffices, by Lemma 4.14, to verify that

$$y'_{si_1 \dots si_{k-1} si_k} = (g_{i_1 \dots i_{k-1}}(j \mapsto y'_{i_1 \dots i_{k-1} j}))(si_k) \qquad \text{for all } i_k : \mathbf{I}.$$

But this follows by

$$\begin{aligned} y'_{si_1 \dots si_{k-1} si_k} &= f_{i_1 \dots i_k}(y'_{i_1 \dots i_k}) && \text{by (9′)} \\ &= (g_{i_1 \dots i_{k-1}}(y'_{i_1 \dots i_{k-1}(-)}))(si_k) && \text{by (16).} \end{aligned}$$

□

5 The min and $\bigsqcup$ operations

In this section, we define a binary "minimum" operation, min, on $\mathbf{I}$, and use it to characterize a "limit-finding" operator, $\bigsqcup^X$, on any complete object. Both operations will be used extensively in the sequel.

Using Proposition 4.4, define $\min : \mathbf{I} \times \mathbf{I} \longrightarrow \mathbf{I}$ to be the unique bistrict map such that $\min(si, sj) = s(\min(i, j))$.

Lemma 5.1 min *is a binary semilattice structure on* $\mathbf{I}$*, i.e.*

$$\min(i, i) = i \tag{17}$$
$$\min(i, j) = \min(j, i) \tag{18}$$
$$\min(i, \min(j, k)) = \min(\min(i, j), k). \tag{19}$$

PROOF. For (17), the map $i \mapsto \min(i, i)$ is strict, by Proposition 4.3, and satisfies $\min(si, si) = s(\min(i, i))$. Thus $(i \mapsto \min(i, i)) = \mathsf{id}_{\mathbf{I}}$, by the $k = 1$ case of Proposition 4.4.

For (18), the maps $(i, j) \mapsto \min(i, j)$ and $(i, j) \mapsto \min(j, i)$ are bistrict and satisfy $\min(si, sj) = s(\min(i, j))$ and $\min(sj, si) = s(\min(j, i))$ respectively. So, by the $k = 2$ case of Proposition 4.4, they are equal.

For (19), the maps $(i,j,k) \mapsto \min(i, \min(j,k))$ and $(i,j,k) \mapsto \min(\min(i,j),k)$ are 3-strict and respectively satisfy $\min(si, \min(sj, sk)) = s(\min(i, \min(j,k))$ and $\min(\min(si, sj), sk) = s(\min(\min(i,j),k))$. So, by the $k = 3$ case of Proposition 4.4, they are equal. $\Box$

Our first application of min is to characterize the unique "limit-finding" operator $\bigsqcup^X : X^{\mathbf{I}} \longrightarrow X$, on any complete object X. To manipulate expressions involving $\bigsqcup^X$, we often omit the superscript X, and we use the convenient notation $\bigsqcup_j x_j$ for $\bigsqcup(j \mapsto x_j)$.

Proposition 5.2 *If X is complete then:*

(1) There exists a unique map $\bigsqcup^X : X^{\mathbf{I}} \longrightarrow X$ satisfying

$$x_i = \bigsqcup_j x_{\min(i,j)} \qquad \text{for all } x_{(-)} : X^{\mathbf{I}} \text{ and } i : \mathbf{I}. \tag{20}$$

(2) $\bigsqcup^X$ satisfies

$$\bigsqcup_i x = x, \qquad \text{for all } x : X \tag{21}$$

$$\bigsqcup_i x_i = \bigsqcup_i x_{si}, \qquad \text{for all } x_{(-)} : X^{\mathbf{I}} \tag{22}$$

$$\bigsqcup_i (\bigsqcup_j x_{ij}) = \bigsqcup_i x_{ii} \qquad \text{for all } x_{(-)(-)} : X^{\mathbf{I}\times\mathbf{I}}. \tag{23}$$

(3) For any $f : X \longrightarrow Y$, where Y is complete,

$$f(\bigsqcup_i^X x_i) = \bigsqcup_i^Y f(x_i) \qquad \text{for all } x_{(-)} : X^{\mathbf{I}}. \tag{24}$$

(4) A subobject $Z \longmapsto X$ is complete if and only if

$$\bigsqcup_i^X z_i \in Z \qquad \text{for all } z_{(-)} : Z^{\mathbf{I}}.$$

The properties in statement (2) are those of a *formal lub operator* [6]. Statement (3) expresses that all functions between complete sets are "continuous" in a natural sense. Property (4) will be useful in Section 13.

To prove Proposition 5.2, it is convenient to develop additional properties of the initial $\mathbb{L}$-algebra $(\mathbf{I}, \sigma)$ and final $\mathbb{L}$-coalgebra $(\mathbf{F}, \tau)$, relating to the canonical map $\iota : \mathbf{I} \longrightarrow \mathbf{F}$. Define $\infty : \mathbf{F}$ to be the point given by the unique coalgebra homomorphism $\infty : \mathbf{1} \longrightarrow \mathbf{F}$ from $\{\cdot\} : \mathbf{1} \longrightarrow \mathbb{L}\mathbf{1}$ to τ (this agrees with the explicit definition $\infty = (n \mapsto \top)$ in Section 2).

Lemma 5.3 *There exists a morphism* $s' \colon \mathbf{F} \longrightarrow \mathbf{F}$ *satisfying:*

$$s'(\iota(i)) \;=\; \iota(s(i)) \qquad \text{for all } i \colon \mathbf{I} \tag{25}$$

$$s'(\infty) \;=\; \infty \tag{26}$$

PROOF. Define $s' = \tau^{-1} \circ \{\cdot\}$. Then, by the definition of ι, we have $s' \circ \iota = \tau^{-1} \circ \{\cdot\} \circ \iota = \tau^{-1} \circ \mathbb{L}\iota \circ \{\cdot\} = \iota \circ \sigma \circ \{\cdot\} = \iota \circ s$. Similarly, by the definition of ∞ we have, $s' \circ \infty = \tau^{-1} \circ \{\cdot\} \circ \infty = \tau^{-1} \circ \mathbb{L}\infty \circ \{\cdot\} = \tau^{-1} \circ \tau \circ \infty = \infty$. $\square$

Lemma 5.4 *There exists a morphism* $\min' \colon \mathbf{I} \times \mathbf{F} \longrightarrow \mathbf{I}$ *satisfying:*

$$\min'(i, \iota(j)) \;=\; \min(i, j) \qquad \text{for all } i, j \colon \mathbf{I} \tag{27}$$

$$\min'(i, \infty) \;=\; i \qquad \text{for all } i \colon \mathbf{I} \tag{28}$$

PROOF. Just as $\sigma \circ \bigcup \circ \mathbb{L}\sigma^{-1}$ is a pointed structure on $\mathbf{I}$, so is $\tau^{-1} \circ \bigcup \circ \mathbb{L}\tau$ a pointed structure on $\mathbf{F}$. Consider the pointed object $\mathbf{F} \multimap \mathbf{I}$ of strict functions, defined in the text following Lemma 4.12. For any strict $f \colon \mathbf{F} \to \mathbf{I}$, the composite $\sigma \circ \mathbb{L}f \circ \tau$ is strict, as all its components are. Thus we have a morphism $\sigma \circ \mathbb{L}(-) \circ \tau \colon (\mathbf{F} \multimap \mathbf{I}) \longrightarrow (\mathbf{F} \multimap \mathbf{I})$. Let $\min'' \colon \mathbf{I} \longrightarrow (\mathbf{F} \multimap \mathbf{I})$ be the unique strict morphism such that $\min''(s(i)) = \sigma \circ \mathbb{L}(\min''(i)) \circ \tau$, as given by Proposition 4.4. Define $\min' \colon \mathbf{I} \times \mathbf{F} \longrightarrow \mathbf{I}$ by $\min'(i, j') = \min''(i)(j')$. By Lemma 4.13, $\min'$ is bistrict. Moreover, for any $i \colon \mathbf{I}$,

$$\min''(s(i)) \circ s' \;=\; \sigma \circ \mathbb{L}(\min''(i)) \circ \{\cdot\} \;=\; \sigma \circ \{\cdot\} \circ \min''(i) \;=\; s \circ \min''(i).$$

Thus $\min'(s(i), s'(j')) = s(\min'(i, j'))$.

As $\iota \colon \mathbf{I} \longrightarrow \mathbf{F}$ is strict, we have that $(i, j) \mapsto \min'(i, \iota(j)) \colon \mathbf{I} \times \mathbf{I} \longrightarrow \mathbf{I}$ is bistrict and satisfies $\min'(s(i), \iota(s(j))) = \min'(s(i), s'(\iota(j))) = s(\min'(i, j'))$. Thus, by the characterizing property of min, indeed $\min'(i, \iota(j)) = \min(i, j)$.

For equation (28), we have that $i \mapsto \min'(i, \infty) \colon \mathbf{I} \longrightarrow \mathbf{I}$ is strict, and satisfies $\min'(s(i), \infty) = \min'(s(i), s'(\infty)) = s(\min'(i, \infty))$. By Proposition 4.4, the identity is the only such map. Thus indeed $\min'(i, \infty) = i$. $\square$

PROOF of Proposition 5.2. For statement (1), we first define $\bigsqcup^X$. Given $x_{(-)} \colon X^{\mathbf{I}}$, there exists, by the definition of completeness, a unique $x'_{(-)} \colon X^{\mathbf{F}}$ satisfying $x'_{\iota(i)} = x_i$, for all $x \colon \mathbf{I}$. Define $\bigsqcup_i x_i = x'_\infty$.

To show (20), take any $i \colon \mathbf{I}$. Consider $x_{\min(i,-)} \colon X^{\mathbf{I}}$. Then $x'_{j'} = x_{\min'(i,j')}$ defines $x'_{(-)} \colon X^{\mathbf{F}}$ satisfying $x'_{\iota(j)} = x_{\min(i,j)}$, by (27). Thus indeed

$$\bigsqcup_j x_{\min(i,j)} \;=\; x'_\infty \;=\; x_{\min'(i,\infty)} \;=\; x_i \,,$$

with the last equality given by (28).

For uniqueness, suppose that $\bigsqcup' \colon X^{\mathbf{I}} \longrightarrow \mathbf{I}$ satisfies $\bigsqcup'_j x_{\min(i,j)} = x_i$. Define $x'_{(-)} \colon X^{\mathbf{F}}$ by $x'_{i'} = \bigsqcup'_j x_{\min'(j,i')}$. Then

$$x'_{\iota(i)} \;=\; \bigsqcup_j{}' x_{\min'(j,\iota(i))} \;=\; \bigsqcup_j{}' x_{\min(j,i)} \;=\; x_i.$$

So, by the completeness of X, we have that $x'_{(-)}$ is the unique element of $X^{\mathbf{F}}$ satisfying $x'_{\iota(i)} = x_i$. So, by the definitions of $\bigsqcup$ and $x'_{(-)}$,

$$\bigsqcup_i x_i \;=\; x'_\infty \;=\; \bigsqcup_i{}' x_{\min'(i,\infty)} \;=\; \bigsqcup_i{}' x_i.$$

For statement (2), equation (21) holds because the constant function $k_x = (i \mapsto x) \colon X^{\mathbf{I}}$ extends to $k'_x = (i' \mapsto x) \colon X^{\mathbf{F}}$, so $\bigsqcup_i x = k'_x(\infty) = x$.

For (22), given $x_{(-)} \colon X^{\mathbf{I}}$, let $x'_{(-)} \colon X^{\mathbf{F}}$ be the unique family such that $x'_{\iota(i)} = x_i$. Then $x'_{s'(-)} \colon X^{\mathbf{F}}$ extends $x_{s(-)} \colon X^{\mathbf{I}}$, because, by (25), we have $x_{s(i)} = x'_{\iota(s(i))} = x'_{s'(\iota(i))}$. Thus, by (26), we have $\bigsqcup_i x_{s(i)} = x'_{s'(\infty)} = x'_\infty = \bigsqcup_i x_i$.

For (23), consider $x_{(-)(-)} \colon X^{\mathbf{I}\times\mathbf{I}}$. For any $i \colon \mathbf{I}$, define $y_{i\,(-)} \colon X^{\mathbf{F}}$ as the unique family satisfying $y_{i\,\iota(j)} = x_{ij}$. Thus $\bigsqcup_j x_{ij} = y_{i\,\infty}$. Also, for each $j' \colon \mathbf{F}$, define $z_{(-)\,j'} \colon X^{\mathbf{F}}$ as the unique family satisfying $z_{\iota(i)\,j'} = y_{ij'}$. Then $\bigsqcup_i(\bigsqcup_j x_{ij}) = \bigsqcup_i y_{i\,\infty} = z_{\infty\,\infty}$. On the other hand, consider $i' \mapsto z_{i'\,i'} \colon X^{\mathbf{F}}$. This satisfies $z_{\iota(i)\,\iota(i)} = y_{i\,\iota(i)} = x_{ii}$. So $\bigsqcup_i x_{ii} = z_{\infty\,\infty}$. Thus indeed $\bigsqcup_i(\bigsqcup_j x_{ij}) = \bigsqcup_i x_{ii}$.

For statement (3), consider any $x_{(-)} \colon X^{\mathbf{I}}$. By the completeness of X, let $x'_{(-)} \colon X^{\mathbf{F}}$ be the unique such that $x_{\iota(i)} = x_i$. Then, by the completeness of Y, we have that $f(x'_{(-)})$ is the unique $y'_{(-)} \colon Y^{\mathbf{F}}$ satisfying $y_{\iota(i)} = f(x_i)$. So by the definitions of $\bigsqcup^X$ and $\bigsqcup^Y$, we have $f(\bigsqcup^X_i x_i) = f(x'_\infty) = \bigsqcup^Y_i f(x_i)$ as required.

Finally, for statement (4), we have a subobject $m \colon Z \longrightarrow\!\!\!\!\!\rightarrow X$. First, suppose $\bigsqcup^X_i z_i \in Z$ for all $z_{(-)} \colon Z^{\mathbf{I}}$. Consider any $z_{(-)} \colon Z^{\mathbf{I}}$. By the assumption, we can define $z'_{(-)} \colon Z^{\mathbf{F}}$ by $z'_{i'} = \bigsqcup^X_j z_{\min'(j,i')}$. This satisfies $z'_{\iota(i)} = z_i$ for all i. It is the unique such because, as X is complete and m is mono, $z'_{(-)}$ is determined by $m(z'_{(-)}) \colon X^{\mathbf{F}}$ being unique such that $m(z'_{\iota(i)}) = m(z_i)$ for all i. Thus Z is indeed complete. Conversely, suppose that Z is complete and consider any $z_{(-)} \colon Z^{\mathbf{I}}$. Let $z'_{(-)} \colon Z^{\mathbf{F}}$ be the unique function such that $z'_{\iota(i)} = z_i$ for all i. As X is complete, we have that $m(z'_{(-)}) \colon X^{\mathbf{F}}$ is the unique map satisfying $m(z'_{\iota(i)}) = m(z_i)$ for all i. Thus, by the definition of $\bigsqcup^X$, we have $\bigsqcup^X_i(m(z_i)) = m(z'_\infty)$, i.e. indeed $\bigsqcup^X_i z_i \in Z$. $\square$

6 The limit-colimit coincidence

One of the main tools in the proof of Theorem 1 will be a variant of the limit-colimit coincidence of domain theory. The standard domain-theoretic version of this coincidence uses **N**-indexed diagrams of *embedding-projection (e-p)* pairs, see e.g. [45]. In our setting, there is no obvious notion of e-p pair to use. A similar issue was addressed by Plotkin, who developed a generalized notion of e-p pair to establish a limit-colimit coincidence in the context of axiomatic domain theory [34]. However, as motivated in Section 3, we have to depart from the standard theorem in another significant way: our diagrams must be indexed by **I** not **N**. Miraculously, the use of **I** as an indexing object enables us to do away with the notion of e-p pair entirely. Instead we prove a limit-colimit coincidence for arbitrary diagrams satisfying two simple equational properties.

Let K be an internal category in **C**. For this entire section, we reason internally in **C** about K. As we do not require K to be locally small, we refer to $\{\mathsf{K}(A,B)\}_{A,B\in|\mathsf{K}|}$ as the family of *hom-classes*.

An **I**-*bichain* in K is given by families,

$$A_{(-)}\colon |\mathsf{K}|^{\mathbf{I}} \qquad\qquad x_{(-)(-)}\colon \prod_{i:\mathbf{I}}\prod_{j:\mathbf{I}}\mathsf{K}(A_i,A_j),$$

satisfying the equations

$$x_{ii} = \mathsf{id}_{A_i} \tag{29}$$

$$x_{jk}\circ x_{ij} = x_{\min(i,j,k)\,k}\circ x_{i\,\min(i,j,k)}. \tag{30}$$

Here $\min(i,j,k)$ means $\min(i,\min(j,k))$ (equivalently $\min(\min(i,j),k)$, by Lemma 5.1), using the min operation from Section 4.

Henceforth, we shall use the semilattice properties of min, given by Lemma 5.1, without further reference, and we shall make free use of evident derived operations such as $\min(i,j,k)$ above. We shall also use an internal partial order on **I**, derived from the semilattice structure in the standard way:

$$i \sqsubseteq j \;\text{ iff }\; i = \min(i,j).$$

Equations (29) and (30) for an **I**-bichain have useful consequences relating $x_{(-)(-)}$ to the partial order $\sqsubseteq$ on **I**.

Lemma 6.1 *For any $i,j,k:\mathbf{I}$,*

$$\textit{if } i\sqsubseteq j \textit{ or } k\sqsubseteq j \textit{ then } x_{jk}\circ x_{ij} = x_{ik}. \tag{31}$$

In particular, if $i\sqsubseteq j$ then $x_{ji}\circ x_{ij} = \mathsf{id}_{A_i}$.

PROOF. Suppose $i \sqsubseteq j$. Then:

$$\begin{aligned} x_{jk} \circ x_{ij} &= x_{\min(i,j,k)\,k} \circ x_{i\,\min(i,j,k)} && \text{by (30)} \\ &= x_{\min(i,k)\,k} \circ x_{i\,\min(i,k)} && \text{as } i \sqsubseteq j \\ &= x_{i\,k} \circ x_{i\,i} && \text{by (30)} \\ &= x_{i\,k} && \text{by (29).} \end{aligned}$$

The equation $x_{ji} \circ x_{ij} = \mathsf{id}_{A_i}$ follows, by (29). The argument for $k \sqsubseteq j$ is similar. $\Box$

Thus if $i \sqsubseteq j$ then x_{ij} and x_{ji} form a section-retraction pair. The limit-colimit coincidence relates the colimit of the diagram of sections to the limit of the diagram of retractions.

Given an $\mathbf{I}$-bichain, $(A_{(-)}, x_{(-)(-)})$, we write $(x_{ij})_{i \sqsubseteq j}$ for the evident partially-ordered diagram of shape $(\mathbf{I}, \sqsubseteq)$, consisting entirely of sections. As usual, a *cocone* for $(x_{ij})_{i \sqsubseteq j}$ is given by an object B of K together with $c_{(-)} : \prod_{i:\mathbf{I}} \mathsf{K}(A_i, B)$ such that, for all $i \sqsubseteq j : \mathbf{I}$, it holds that $c_i = c_j \circ x_{ij}$. A *mediating morphism* from $(B, c_{(-)})$ to another cocone $(B', c'_{(-)})$ is a morphism $b\colon B \longrightarrow B'$ in K such that, for all $i : \mathbf{I}$, it holds that $c'_i = b \circ c_i$. The cocone $(B, c_{(-)})$ is *colimiting* if, for any cocone $(B', c'_{(-)})$, there exists a unique mediating morphism from $(B, c_{(-)})$ to $(B', c'_{(-)})$.

Dually, we write $(x_{ij})_{i \sqsupseteq j}$ for the evident partially-ordered diagram of shape $(\mathbf{I}, \sqsupseteq)$, consisting of retractions. The notion of *cone*, $l_{(-)} : \prod_{i:\mathbf{I}} \mathsf{K}(B, A_i)$, and *limit* are defined in the obvious way.

Proposition 6.2 (Limit-colimit coincidence)
If K is an internal category in which all hom-classes are complete then, for any $\mathbf{I}$-bichain $(A_{(-)}, x_{(-)(-)})$ in K, the following statements are equivalent.

(1) B is a limiting object for $(x_{ij})_{i \sqsupseteq j}$.
(2) There exists a cone $l_{(-)} : \prod_{i:\mathbf{I}} \mathsf{K}(B, A_i)$ for $(x_{ij})_{i \sqsupseteq j}$, and also a cocone $c_{(-)} : \prod_{i:\mathbf{I}} \mathsf{K}(A_i, B)$ for $(x_{ij})_{i \sqsubseteq j}$ such that:

$$l_j \circ c_i = x_{ij} \qquad \text{for all } i, j : \mathbf{I} \tag{32}$$

$$\bigsqcup_i (c_i \circ l_i) = \mathsf{id}_B. \tag{33}$$

(3) B is a colimiting object for $(x_{ij})_{i \sqsubseteq j}$.

Moreover, if (2) holds then $l_{(-)}$ is a limiting cone and $c_{(-)}$ is a colimiting cone. Furthermore, (32) and (33) together imply that each of $l_{(-)}$ and $c_{(-)}$ determines the other.

In view of the proposition, we shall henceforth refer to $(B, l_{(-)}, c_{(-)})$ satisfying (32) and (33) as a *bilimit* of the **I**-bichain $(A_{(-)}, x_{(-)(-)})$.

The remainder of the section is devoted to the proof of Proposition 6.2. Accordingly, assume hom-classes in K are complete and that $(A_{(-)}, x_{(-)(-)})$ is an **I**-bichain in K.

Lemma 6.3 *If $l_{(-)}$ is any cone for $(x_{ij})_{i \sqsupseteq j}$ then $l_j = \bigsqcup_i (x_{ij} \circ l_i)$. Similarly, if $c_{(-)}$ is a cocone for $(x_{ij})_{i \sqsubseteq j}$ then $c_j = \bigsqcup_i (c_i \circ x_{ji})$.*

PROOF. The proofs of both equalities are similar, so we just show the equality for l_j. As l_j is a cone, we have $i \sqsupseteq j$ implies $l_j = x_{ij} \circ l_i$. So:

$$\begin{aligned}
l_j &= \bigsqcup_i l_{\min(i,j)} && \text{by (20)} \\
&= \bigsqcup_i (x_{\,i\,\min(i,j)} \circ l_i) && \text{as } l_{(-)} \text{ is a cone} \\
&= \bigsqcup_i \bigsqcup_{i'} (x_{\,i\,\min(i',j)} \circ l_i) && \text{by (23)} \\
&= \bigsqcup_i (x_{ij} \circ l_i) && \text{by (20).}
\end{aligned}$$

□

Lemma 6.4 *If $l_{(-)}$ and $c_{(-)}$ are a cone and cocone satisfying (32) then, for any cone $l'_{(-)}$, it holds that $\bigsqcup_i (c_i \circ l'_i)$ is a mediating morphism from $l'_{(-)}$ to $l_{(-)}$.*

PROOF. We must show that $l'_j = l_j \circ \bigsqcup_i (c_i \circ l'_i)$. But, by (24), (32) and Lemma 6.3,

$$l_j \circ \bigsqcup_i (c_i \circ l'_i) \;=\; \bigsqcup_i\, l_j \circ c_i \circ l'_i \;=\; \bigsqcup_i\, x_{ij} \circ l'_i \;=\; l'_j.$$

□

PROOF of Proposition 6.2. To prove that (1) implies (2), suppose that $l_{(-)}$ is a limiting cone for $(x_{jk})_{j \sqsupseteq k}$. For any $i : \mathbf{I}$, consider the family of morphisms $\{x_{ij} : A_i \to A_j\}_{j : \mathbf{I}}$. By (31), this is a cone for $(x_{jk})_{j \sqsupseteq k}$. Therefore, as $l_{(-)}$ is limiting, there is a unique morphism $c_i : A_i \to B$ such that, for all j, it holds that $x_{ij} = l_j \circ c_i$. This establishes (32). We must show that $c_{(-)}$ is a cocone for $(x_{ij})_{i \sqsubseteq j}$ and that (33) holds.

For the cocone, we must show that $i \sqsubseteq j$ implies that $c_i = c_j \circ x_{ij}$; for which, by the defining property of c_i, it suffices to show that, $i \sqsubseteq j$ implies that

$x_{ik} = l_k \circ c_j \circ x_{ij}$ for all k. But this holds because, by (31) and (32),

$$x_{ik} \;=\; x_{jk} \circ x_{ij} \;\;=\; l_k \circ c_j \circ x_{ij}.$$

Equation (33) now follows from Lemma 6.4, because the morphism $\bigsqcup_i (c_i \circ l_i)$ mediates from the cone $l_{(-)}$ to itself and id_B is the unique mediating morphism. Thus $\bigsqcup_i (c_i \circ l_i) = \mathsf{id}_B$. This proves that 1 implies 2.

To prove that (2) implies (1), consider any cone $l_{(-)}$ and cocone $c_{(-)}$ satisfying (32) and (33). We show that $l_{(-)}$ is limiting for $(x_{jk})_{j \sqsupseteq k}$. Accordingly, let $l'_{(-)}$ be any other cone. By Lemma 6.4, we have that $\bigsqcup_i (c_i \circ l'_i)$ mediates from $l'_{(-)}$ to $l_{(-)}$. We must show that it is the unique mediating morphism. Suppose then that y also mediates between the two cones, i.e. for all $i : \mathbf{I}$, it holds that $l'_i = l_i \circ y$. Then indeed, using (33) and (24),

$$y = (\bigsqcup_i c_i \circ l_i) \circ y \;\;=\; \bigsqcup_i (c_i \circ l_i \circ y) \;\;=\; \bigsqcup_i c_i \circ l'_i.$$

We have thus shown that statement (1) is equivalent to statement (2). The equivalence of statements (2) and (3) follows by duality.

To complete the proof, we show that (32) and (33) imply that each of $l_{(-)}$ and $c_{(-)}$ determines the other. To see that $l_{(-)}$ determines $c_{(-)}$, suppose that we have $c_{(-)}$ and $c'_{(-)}$, both satisfying (32) and (33) with respect to $l_{(-)}$. We have shown that $l_{(-)}$ is limiting, and that, for $i : \mathbf{I}$, the family $\{x_{ij} : A_i \to A_j\}_{j : \mathbf{I}}$ is a cone for $(x_{jk})_{j \sqsupseteq k}$. Then, by (32), c_i and c'_i are both the unique morphism $c_i : A_i \to B$ from the cone $\{x_{ij} : A_i \to A_j\}_{j : \mathbf{I}}$ to the limiting cone $l_{(-)}$. Thus indeed $c_i = c'_i$. The argument that $c_{(-)}$ determines $l_{(-)}$ is dual. $\square$

7 Conditions for algebraic compactness

In this section we define a notion of *suitable* internal category—one satisfying conditions that are sufficient for algebraic compactness to hold. These conditions are convenient for establishing the algebraic compactness of specific internal categories, e.g. pP.

Definition 7.1 (Suitable category) A *suitable* category is given by an internal category K together with a pointed structure $(|\mathsf{K}|, \alpha)$, on the class of objects, and a family of pointed structures $\{(\mathsf{K}(A,B),\, \beta_{A,B})\}_{A,B : |\mathsf{K}|}$, on the hom-classes, satisfying: for all $A, B, C : |\mathsf{K}|$, the hom-class $\mathsf{K}(A,B)$ is complete and the composition function $\mathsf{K}(B,C) \times \mathsf{K}(A,B) \to \mathsf{K}(A,C)$ is bistrict; $\{\mathsf{id}_A : \mathsf{K}(A,A)\}_{A : |\mathsf{K}|}$ is a strict family; and every $\mathbf{I}$-bichain in K has a specified bilimit.

In this definition, by having a specified bilimit we mean that bilimits are given by a morphism $\mathrm{Bichains}^{\mathsf{K}} \longrightarrow \mathrm{Bicones}^{\mathsf{K}}$ in $\mathbf{C}$, where $\mathrm{Bichains}^{\mathsf{K}}$ is the class of $\mathbf{I}$-bichains in K and $\mathrm{Bicones}^{\mathsf{K}}$ is the class of cone/cocone tuples $(B, l_{(-)}, c_{(-)})$ for $\mathbf{I}$-bichains.

Proposition 7.2 *If* K *and* L *are suitable then so are* K^{op} *and* $\mathsf{K} \times \mathsf{L}$.

PROOF. The result for K^{op} is straightforward, using the duality inherent in the notions of $\mathbf{I}$-bichains and $\mathbf{I}$-bilimits. For $\mathsf{K} \times \mathsf{L}$, the required pointed structures are the products of those of K and L, and the specified bilimits are obtained by pairing the specified bilimits in K and L. □

The next result is the reason for introducing the notion of suitable category.

Proposition 7.3 *Every suitable internal category is algebraically compact.*

The remainder of the section is devoted to the proof of Proposition 7.3. We first show how to construct a bifree algebra for a single internal functor, and then, subsequently, derive the full power of Definition 3.3 for families of internal functors. Accordingly, let F be an endofunctor on a suitable category K.

Because $(|\mathsf{K}|, \alpha)$ is pointed, there is, by Proposition 4.4, a unique strict map $F^{(-)}\mathsf{0} : \mathbf{I} \longrightarrow |\mathsf{K}|$ such that $F(F^i\mathsf{0}) = F^{si}\mathsf{0}$. Here the notation is chosen to convey the idea that one should think of $F^i\mathsf{0}$ as the i-th iterate of F applied to a zero object $\mathsf{0}$ in K. However, this intuition is subject to two caveats: firstly i comes from $\mathbf{I}$ rather than from $\mathbf{N}$, so the notion of iterate is non-standard; secondly, we have not required that K have a zero object (although the existence of one follows from Proposition 7.3, once proven).

As each $(\mathsf{K}(A, B), \beta_{A,B})$ is pointed, there exists, by Proposition 4.9, a unique bistrict family $x_{(-)(-)} \colon \prod_{i:\mathbf{I}} \prod_{j:\mathbf{I}} \mathsf{K}(F^i\mathsf{0}, F^j\mathsf{0})$ satisfying

$$x_{\,si\,sj} = F(x_{ij})\,. \tag{34}$$

Lemma 7.4 $(F^{(-)}\mathsf{0}, x_{(-)(-)})$ *is an* $\mathbf{I}$*-bichain.*

PROOF. For equation (29), define $y_i = x_{ii}$. As $x_{(-)(-)}$ is bistrict, it follows from Proposition 4.8 that $y_{(-)} \colon \prod_{i:\mathbf{I}} \mathsf{K}(F^i\mathsf{0}, F^i\mathsf{0})$ is a strict family. By the $k = 1$ case of Proposition 4.9, $y_{(-)}$ is thus the unique strict family satisfying $y_{si} = F(y_i)$. As $F^{(-)}\mathsf{0} : \mathbf{I} \longrightarrow |\mathsf{K}|$ is strict and $\{\mathsf{id}_A \colon \mathsf{K}(A, A)\}_{A:|\mathsf{K}|}$ is a strict family, the composite $\mathsf{id}_{F^{(-)}\mathsf{0}} \colon \prod_{i:\mathbf{I}} \mathsf{K}(F^i\mathsf{0}, F^i\mathsf{0})$ is also a strict family. Clearly this satisfies $\mathsf{id}_{F^{s(i)}\mathsf{0}} = F(\mathsf{id}_{F^i\mathsf{0}})$. Thus indeed $x_{ii} = y_i = \mathsf{id}_{F^{s(i)}\mathsf{0}}$.

For equation (30), define $y_{(-)\,(-)\,(-)}, z_{(-)\,(-)\,(-)} \colon \prod_{i:\mathbf{I}} \prod_{j:\mathbf{I}} \prod_{k:\mathbf{I}} \mathsf{K}(F^i 0,\, F^k 0)$ by $y_{ijk} = x_{jk} \circ x_{ij}$ and $z_{ijk} = x_{\min(i,j,k)\,k} \circ x_{i\,\min(i,j,k)}$. We show below that $y_{(-)\,(-)\,(-)}$ and $z_{(-)\,(-)\,(-)}$ are 3-strict and satisfy $y_{s(i)\,s(j)\,s(k)} = F(y_{ijk})$ and $z_{s(i)\,s(j)\,s(k)} = F(z_{ijk})$. Hence, by the $k = 3$ case of Proposition 4.9, $y_{ijk} = z_{ijk}$. Thus indeed $x_{jk} \circ x_{ij} = x_{\min(i,j,k)\,k} \circ x_{i\,\min(i,j,k)}$.

We first show that $y_{(-)\,(-)\,(-)}$ is 3-strict. Fix j, k and consider the family $y_{(-)\,j\,k} \colon \prod_{i:\mathbf{I}} \mathsf{K}(F^i 0,\, F^k 0)$. Then $x_{(-)\,j}$ is a strict family, because $x_{(-)\,(-)}$ is bistrict; and $\{x_{jk} \circ (-) \colon \mathsf{K}(F^i 0,\, F^j 0) \to \mathsf{K}(F^i 0,\, F^k 0)\}_{i:\mathbf{I}}$ is a family of strict maps, because composition in K is bistrict. So the composite $x_{jk} \circ x_{(-)\,j} = y_{(-)\,j\,k}$ is a strict family. By a similar argument, $y_{i\,j\,(-)}$ is a strict family, for all i, j. Lastly, fix i, k and consider $y_{i\,(-)\,k}$. We have that $x_{(-)\,k}$ and $x_{i\,(-)}$ are strict families, so the pairing $(x_{(-)\,k}, x_{i\,(-)}) \colon \prod_{j:\mathbf{I}} (\mathsf{K}(F^j 0, F^k 0) \times \mathsf{K}(F^i 0, F^j 0))$ is a strict family. Also, composition in K gives a family of bistrict functions $\{\mathsf{K}(F^j 0,\, F^k 0) \times \mathsf{K}(F^i 0,\, F^j 0) \to \mathsf{K}(F^i 0,\, F^k 0)\}_{j:\mathbf{I}}$, each of which is strict by Proposition 4.3. Therefore the composite family $x_{(-)\,k} \circ x_{i\,(-)} = y_{i\,(-)\,k}$ is strict. Thus $y_{(-)\,(-)\,(-)}$ is indeed 3-strict.

To prove the 3-strictness of $z_{(-)\,(-)\,(-)}$, we show that each of $x_{\min(i,j,k)\,k}$ and $x_{i\,\min(i,j,k)}$ is 3-strict, whence, by the strictness of composition, the composite $x_{\min(i,j,k)\,k} \circ x_{i\,\min(i,j,k)} = z_{ijk}$ is indeed 3-strict. We just give the argument for $x_{\min(i,j,k)\,k}$ as that for $x_{i\,\min(i,j,k)}$ is similar. The strictness of $x_{\min(i,j,k)\,k}$ in i (for fixed j, k) is obvious by the 3-strictness of min and the bistrictness of $x_{(-)\,(-)}$. Strictness in j is by the same argument. For strictness in k, fix i, j. Consider the family $\{x_{\min(i,j,k_1)\,k_2}\}_{k_1:\mathbf{I},\, k_2:\mathbf{I}}$. This is obviously bistrict. By Proposition 4.8, $\{x_{\min(i,j,k_1)\,k_2}\}_{(k_1,k_2):\mathbf{I}\times\mathbf{I}}$ is thus a strict family. Hence, by composing with the strict $k \mapsto (k, k)$, the family $x_{\min(i,j,k)\,k}$ is strict in k.

To show $y_{s(i)\,s(j)\,s(k)} = F(y_{ijk})$, we have

$$\begin{aligned}
y_{s(i)\,s(j)\,s(k)} &= x_{s(j)\,s(k)} \circ x_{s(i)\,s(j)} && \text{def. of } y_{(-)\,(-)\,(-)} \\
&= F(x_{jk}) \circ F(x_{ij}) && \text{by (34)} \\
&= F(x_{jk} \circ x_{ij}) && F \text{ an internal functor} \\
&= F(y_{ijk}) && \text{def. of } y_{(-)\,(-)\,(-)}.
\end{aligned}$$

Finally, to show $z_{s(i)\,s(j)\,s(k)} = F(z_{ijk})$, we have

$$\begin{aligned}
z_{s(i)\,s(j)\,s(k)} &= x_{\min(s(i),s(j),s(k))\,s(k)} \circ x_{s(i)\,\min(s(i),s(j),s(k))} && \text{def. of } z_{(-)\,(-)\,(-)} \\
&= x_{s(\min(i,j,k))\,s(k)} \circ x_{s(i)\,s(\min(i,j,k))} && \text{def. of min} \\
&= F(x_{\min(i,j,k)\,k}) \circ F(x_{i\,\min(i,j,k)}) && \text{by (34)} \\
&= F(x_{\min(i,j,k)\,k} \circ x_{i\,\min(i,j,k)}) && F \text{ an internal functor} \\
&= F(z_{ijk}) && \text{def. of } z_{(-)\,(-)\,(-)}.
\end{aligned}$$

□

Now we are in a position to construct the bifree algebra for F. Accordingly, let $(B, l_{(-)}, c_{(-)})$ be the specified bilimit of $(F^{(-)}\mathsf{0}, x_{(-)(-)})$. Define morphisms: $FB \xrightarrow{b} B$ and $B \xrightarrow{b'} FB$ in K by

$$b = \bigsqcup_i (c_{si} \circ Fl_i) \qquad\qquad b' = \bigsqcup_i (Fc_i \circ l_{si}) \tag{35}$$

Lemma 7.5 *b is an isomorphism with inverse b'.*

PROOF. Using equations (22)–(24), (29), (32) and (33), we have:

$$\begin{aligned} b \circ b' &= (\bigsqcup_i c_{si} \circ Fl_i) \circ (\bigsqcup_j Fc_j \circ l_{sj}) = \bigsqcup_i \bigsqcup_j c_{si} \circ Fl_i \circ Fc_j \circ l_{sj} \\ &= \bigsqcup_i c_{si} \circ Fl_i \circ Fc_i \circ l_{si} = \bigsqcup_i c_{si} \circ F(l_i \circ c_i) \circ l_{si} \\ &= \bigsqcup_i c_{si} \circ l_{si} = \bigsqcup_i c_i \circ l_i = \mathsf{id}_B \end{aligned}$$

$$b' \circ b = \bigsqcup_i F\, c_i \circ l_{si} \circ c_{si} \circ Fl_i = \bigsqcup_i F(c_i \circ l_i) = F(\bigsqcup_i c_i \circ l_i) = \mathsf{id}_{FB}.$$

□

We next show that b is the desired bifree F-algebra, by showing that it satisfies a condition equivalent to bifreeness. Suppose that $FA \xrightarrow{a} A$ is an isomorphism in K. Applying Proposition 4.4, define $z_{(-)} : \mathbf{I} \to \mathsf{K}(A, A)$ to be the unique strict function such that the diagram below commutes.

$$\begin{array}{ccc} \mathbf{I} & \xrightarrow{z_{(-)}} & \mathsf{K}(A,A) \\ {\scriptstyle s}\downarrow & & \downarrow{\scriptstyle a \circ F(-) \circ a^{-1}} \\ \mathbf{I} & \xrightarrow[z_{(-)}]{} & \mathsf{K}(A,A) \end{array}$$

We say that a is *special F-invariant* if $\bigsqcup_i z_i = \mathsf{id}_A$.

The notion of special-invariant object was first introduced for cpo-enriched categories in [8]. A generalisation to an axiomatic setting appears in [39], from where the following result is taken. For completeness, we include a proof.

Lemma 7.6 *For any isomorphism $FA \xrightarrow{a} A$, the following are equivalent:*

(1) $FA \xrightarrow{a} A$ is special F-invariant.

(2) $FA \xrightarrow{a} A$ *is an initial* F*-algebra.*
(3) $A \xrightarrow{a^{-1}} FA$ *is a final* F*-coalgebra.*

PROOF. Given any F-algebra, $c\colon FC \longrightarrow C$, define $y^c_{(-)}\colon \mathbf{I} \longrightarrow \mathsf{K}(A,C)$ to be the unique strict map satisfying $y^c_{s(i)} = c \circ F(y^c_i) \circ a^{-1}$, given by Proposition 4.4. In particular, $y^a_i = z_i$. Then $\bigsqcup_i y^c_i$ is an algebra homomorphism from (A,a) to (C,c) because, using (22),

$$(\bigsqcup_i y^c_i) \circ a \;=\; (\bigsqcup_i y^c_{s(i)}) \circ a \;=\; c \circ F(\bigsqcup_i y^c_i) \circ a^{-1} \circ a \;=\; c \circ F(\bigsqcup_i y^c_i).$$

Thus, $\bigsqcup_i z_i$ is an algebra homomorphism from a to a. So when a is the initial algebra, $\bigsqcup_i z_i = \mathsf{id}_A$. Thus indeed (2) implies (1).

For the converse, we must show that the homomorphism $\bigsqcup_i y^c_i$ constructed above is unique. Accordingly, let $x\colon A \longrightarrow C$ in K be any homomorphism, i.e. $c \circ Fx = x \circ a$. Then, by an easy calculation, the right-hand square below commutes. The left-hand square commutes by the definition of $z_{(-)}$.

$$\begin{array}{ccccc}
\mathbf{I} & \xrightarrow{z_{(-)}} & \mathsf{K}(A,A) & \xrightarrow{x \circ (-)} & \mathsf{K}(A,C) \\
{\scriptstyle s}\downarrow & & \downarrow{\scriptstyle a \circ F(-) \circ a^{-1}} & & \downarrow{\scriptstyle c \circ F(-) \circ a^{-1}} \\
\mathbf{I} & \xrightarrow[z_{(-)}]{} & \mathsf{K}(A,A) & \xrightarrow[x \circ (-)]{} & \mathsf{K}(A,C)
\end{array}$$

By the bistrictness of composition $\mathsf{K}(A,C) \times \mathsf{K}(A,A) \to \mathsf{K}(A,C)$, the map $x \circ (-)$ is strict. Hence $x \circ z_{(-)}\colon \mathbf{I} \longrightarrow \mathsf{K}(A,C)$ is a strict map satisfying the defining property of $y^c_{(-)}$, by whose uniqueness $y^c_i = x \circ z_i$. Thus indeed, as a is special F-invariant, $x \;=\; x \circ \bigsqcup_i z_i \;=\; \bigsqcup_i (x \circ z_i) \;=\; \bigsqcup_i y^c_i$.

The equivalence of statements (1) and (3) follows by duality. □

Lemma 7.7 *The isomorphism* $FB \xrightarrow{b} B$ *is special* F*-invariant.*

PROOF. Consider the diagram below.

$$\begin{array}{ccc}
\mathbf{I} & \xrightarrow{c_{(-)} \circ l_{(-)}} & \mathsf{K}(B,B) \\
{\scriptstyle s}\downarrow & & \downarrow{\scriptstyle b \circ F(-) \circ b'} \\
\mathbf{I} & \xrightarrow{c_{(-)} \circ l_{(-)}} & \mathsf{K}(B,B)
\end{array}$$

By (33), we have that $\bigsqcup_i c_i \circ l_i = \mathsf{id}_B$. Thus, for b to be special F-invariant, it suffices to show that $c_{(-)} \circ l_{(-)}$ is strict and makes the diagram above commute.

For strictness, the family $\{c_i : \mathsf{K}(A_i, B)\}_{i:I}$ is obtained as the composite of $\{\mathsf{id}_{A_i} : \mathsf{K}(A_i, A_i)\}_{i:\mathbf{I}}$ with the family $\{c_i \circ (-) : \mathsf{K}(A_i, A_i) \longrightarrow\!\!\!\!\!\rightarrow \mathsf{K}(A_i, B_)\}_{i:\mathbf{I}}$. The former is a strict family because K is suitable, the latter is a family of strict functions because composition in K is bistrict. Thus the composite $\{c_i : \mathsf{K}(A_i, B)\}_{i:I}$ is a strict family. By a similar argument $\{l_i : \mathsf{K}(B, A_i)\}_{i:I}$ is a strict family. Thus $\{(c_i, l_i) : \mathsf{K}(A_i, B) \times \mathsf{K}(B, A_i)\}_{i:\mathbf{I}}$ is also a strict family. By composing this with the family $\{\mathsf{K}(A_i, B) \times \mathsf{K}(B, A_i) \to \mathsf{K}(B, B)\}_{i:\mathbf{I}}$ of composition functions, each of which is strict by Proposition 4.3, we obtain that $\{c_i \circ l_i : \mathsf{K}(B, B)\}_{i:I}$ is a strict family. In other words, $c_{(-)} \circ l_{(-)} : \mathbf{I} \longrightarrow\!\!\!\!\!\rightarrow \mathsf{K}(B, B)$ is a strict map.

For commutativity, we have:

$$
\begin{aligned}
& b \circ F(c_i \circ l_i) \circ b' \\
&\quad = (\bigsqcup_j c_{sj} \circ F l_j) \circ F(c_i \circ l_i) \circ (\bigsqcup_j F c_j \circ l_{sj}) \\
&\quad = \bigsqcup_j c_{sj} \circ F(l_j \circ c_i) \circ F(l_i \circ c_j) \circ l_{sj} && \text{by Proposition 5.2} \\
&\quad = \bigsqcup_j c_{sj} \circ F x_{ij} \circ F x_{ji} \circ l_{sj} && \text{by (32)} \\
&\quad = \bigsqcup_j c_{sj} \circ x_{si\ sj} \circ x_{sj\ si} \circ l_{sj} && \text{by (34)} \\
&\quad = \bigsqcup_j (c_{sj} \circ x_{si\ sj}) \circ \bigsqcup_j (x_{sj\ si} \circ l_{sj}) && \text{by Proposition 5.2} \\
&\quad = \bigsqcup_j (c_j \circ x_{si\ j}) \circ \bigsqcup_j (x_{j\ si} \circ l_j) && \text{by (22)} \\
&\quad = c_{si} \circ l_{si} && \text{by Lemma 6.3.}
\end{aligned}
$$

Thus b is indeed special F-invariant. $\square$

At this point, we have, by Lemmas 7.6 and 7.7, that b is a bifree algebra. Thus we have constructed a bifree algebra for the given internal functor F.

To complete the proof of Proposition 7.3, we must construct bifree algebras for internal families of internal endofunctors, and we must show that these are preserved by reindexing, as in Definition 3.3. The construction part is given by a straightforward relativization of the proof above. Let $\{F_i : \mathsf{K} \to \mathsf{K}\}_{i:I}$ be an internal family of internal functors on K. Then this gives rise to a single internal functor on the internal category $I^*(\mathsf{K})$ in the slice $\mathbf{C}/I$, obtained from K by reindexing. However, the definition of suitable category is plainly preserved by reindexing, thus $I^*(\mathsf{K})$ is a suitable internal category in $\mathbf{C}/I$. The construction given above, when carried out in $\mathbf{C}/I$, thus produces a bifree

algebra for the endofunctor $\{F_i \colon \mathsf{K} \to \mathsf{K}\}_{i:I}$ on $I^*(\mathsf{K})$. This unwinds to give a family $\{a_i : \mathsf{K}(F_i A_i, A_i)\}_{i:I}$ of bifree algebras in K, as required by Definition 3.3.

It remains to show that the constructed families of bifree algebras are preserved by reindexing. For this, we use the following observation. Given a family $\{F_i \colon \mathsf{K} \to \mathsf{K}\}_{i:I}$, where I is an arbitrary index object, the family $\{a_i : \mathsf{K}(F_i A_i, A_i)\}_{i:I}$ constructed above has the following internal characterization. For each $i : I$, the morphism $a_i : \mathsf{K}(F_i A_i, A_i)$ is uniquely determined by its construction using the specified bilimit of the **I**-bichain $(F_i^{(-)}\mathsf{0},\, x_{i(-)(-)})$. Moreover the **I**-bichain is determined by F_i. Thus, internally in **C**, it holds that, for all $i : I$, the morphism a_i equals a map determined uniquely by F_i and the suitability structure on K. This characterization is trivially preserved by reindexing. Thus the constructed families of bifree algebras are preserved by reindexing. This completes the proof of Proposition 7.3.

8 Properties of suitable categories

In this section we establish useful properties of the process of constructing bifree algebras in suitable categories, making use of the constructions of Section 7. The results in this section will be used heavily in Section 12.

The suitablility of a category K is witnessed by the following data in **C**: the pointed structure $(|\mathsf{K}|, \alpha^{\mathsf{K}})$, the family $\{(\mathsf{K}(A,B),\, \beta^{\mathsf{K}}_{A,B})\}_{A,B:|\mathsf{K}|}$ of pointed structures, and the bilimit-finding morphism $\mathrm{bilim}^{\mathsf{K}} \colon \mathrm{Bichains}^{\mathsf{K}} \longrightarrow \mathrm{Bicones}^{\mathsf{K}}$. We henceforth consider a suitable categoriy as being given by a 4-tuple specifying this data, $(\mathsf{K},\, \alpha^{\mathsf{K}},\, \{\beta^{\mathsf{K}}_{A,B})\}_{A,B:|\mathsf{K}|},\, \mathrm{bilim}^{\mathsf{K}})$, although we shall normally elide the additional structure, just writing K.

The data $(\mathsf{K},\, \alpha^{\mathsf{K}},\, \{\beta^{\mathsf{K}}_{A,B})\}_{A,B:|\mathsf{K}|},\, \mathrm{bilim}^{\mathsf{K}})$ determines the construction of bifree algebras in the proof of Proposition 7.3. For any internal functor $F \colon \mathsf{K} \to \mathsf{K}$, we refer to the bifree algebra, $FA \xrightarrow{a} A$, constructed for it as the *canonical* bifree algebra for F. Also, given a functor $G \colon \mathsf{L} \times \mathsf{K} \to \mathsf{K}$, where L is any internal category, we write $G^\dagger \colon \mathsf{L} \to \mathsf{K}$ for the functor, constructed by Proposition 3.4, that finds canonical bifree algebras parametrically.

Any internal functor $F \colon \mathsf{K} \to \mathsf{L}$, between arbitrary internal categories, preserves **I**-bichains, i.e. if $(A_{(-)}, x_{(-)(-)})$ is an **I**-bichain in L then $(FA_{(-)}, Fx_{(-)(-)})$ is an **I**-bichain in L. Similarly, it preserves cones and cocones, hence bicones. Furthermore, by (32) and (33), when hom-classes in K and L are complete, F preserves bilimits of bichains: i.e. if $(B, l_{(-)}, c_{(-)})$ is a bilimit for $(A_{(-)}, x_{(-)(-)})$ then $(FB, Fl_{(-)}, Fc_{(-)})$ is a bilimit for $(FA_{(-)}, Fx_{(-)(-)})$. On the other hand, when K and L are suitable, there is no reason for F to map specified bilimits to specified bilimits. It is useful to identify a strict notion of functor between

suitable categories that does preserve all specified structure up to equality.

Definition 8.1 (Suitable functor) An internal functor $F\colon \mathsf{K} \to \mathsf{L}$, between suitable categories, is *suitable* if: $F\colon (|\mathsf{K}|, \alpha^{\mathsf{K}}) \longrightarrow \mathbb{L}(|\mathsf{L}|, \alpha^{\mathsf{L}})$ is strict; $F_{AB}\colon (\mathsf{K}(A,B), \beta^{\mathsf{K}}_{AB}) \longrightarrow (\mathsf{L}(FA,FB), \beta^{\mathsf{L}}_{FA\,FB})$ is strict, for all $A, B\colon |\mathsf{K}|$; and the diagram below commutes in $\mathbf{C}$,

$$\begin{array}{ccc} \mathrm{Bichains}^{\mathsf{K}} & \xrightarrow{F} & \mathrm{Bichains}^{\mathsf{L}} \\ \Big\downarrow{\scriptstyle \mathrm{bilim}^{\mathsf{K}}} & & \Big\downarrow{\scriptstyle \mathrm{bilim}^{\mathsf{L}}} \\ \mathrm{Bicones}^{\mathsf{K}} & \xrightarrow[F]{} & \mathrm{Bicones}^{\mathsf{L}}\,, \end{array}$$

using the preservation of bicones and bichains discussed above.

The benefit of requiring suitable functors to preserve structure up to equality is that it allows one to avoid coherence conditions that would otherwise arise when comparing, e.g., bifree algebras, between suitable categories. The lemmas below develop several such results.

Lemma 8.2 (Uniformity) *Suppose* K *and* K' *are suitable internal categories, and there is a commuting diagram of internal functors,*

$$\begin{array}{ccc} \mathsf{K} & \xrightarrow{F} & \mathsf{K} \\ \Big\downarrow{\scriptstyle H} & & \Big\downarrow{\scriptstyle H} \\ \mathsf{K}' & \xrightarrow[G]{} & \mathsf{K}'\,, \end{array}$$

with H *suitable. Let* $FA \xrightarrow{a} A$ *be the canonical bifree algebra of* F*, and let* $GB \xrightarrow{b} B$ *be the canonical bifree algebra of* G*. Then* $B = HA$ *and* $b = Ha$*.*

PROOF. By following through the explicit construction of bifree algebras given in Section 7. Specifically, one verifies easily that $G^{(-)}\mathsf{0}\colon \mathbf{I} \longrightarrow |\mathsf{K}'|$ is given by $H \circ F^{(-)}\mathsf{0}$ and that the associated bichain x^G_{ij} is equal to $H(x^F_{ij})$. The canonical bifree algebra of F is given as the bilimit of $(F^{(-)}\mathsf{0}, x^F_{(-)(-)})$, with the algebra map $FA \xrightarrow{a} A$ defined (as b) in (35). The canonical bifree algebra $GB \xrightarrow{b} B$ of G is defined similarly, with B the the bilimit of $(G^{(-)}\mathsf{0}, x^G_{(-)(-)})$. That $B = HA$ follows from H preserving specified bilimits. That $b = Ha$ follows from the definition of these maps using (35). $\Box$

Lemma 8.3 (Parametrized uniformity) *Suppose* K *and* K' *are suitable internal categories, and there is a commuting diagram of internal functors,*

$$\begin{array}{ccc} \mathsf{L} \times \mathsf{K} & \xrightarrow{F} & \mathsf{K} \\ {\scriptstyle J \times H}\downarrow & & \downarrow{\scriptstyle H} \\ \mathsf{L}' \times \mathsf{K}' & \xrightarrow[G]{} & \mathsf{K}', \end{array}$$

with H *suitable. Then* $G^\dagger \circ J = H \circ F^\dagger \colon \mathsf{L} \to \mathsf{K}$.

PROOF. It is enough to prove the result in the case $\mathsf{L}' = \mathsf{L}$ and with J the identity functor, as the general case then follows by an easy application of Proposition 3.5. Accordingly, suppose we have internal functors $F \colon \mathsf{L} \times \mathsf{K} \to \mathsf{K}$ and $G \colon \mathsf{L} \times \mathsf{K}' \to \mathsf{K}'$ and suitable $H \colon \mathsf{K} \to \mathsf{K}'$. We must show that $G^\dagger = H \circ F^\dagger$. For any object C of L, write a_C for the canonical bifree algebra of $F(C, -) \colon \mathsf{K} \to \mathsf{K}$, and b_C for the canonical bifree algebra of $G(C, -) \colon \mathsf{K}' \to \mathsf{K}'$. By Lemma 8.3, $b_C = H(a_C)$. Thus $H \circ F^\dagger$ satisfies the characterizing property of $G^\dagger$, as stated in Proposition 3.4. Thus indeed $G^\dagger = H \circ F^\dagger$. □

Canonical bifree algebras are also preserved by taking opposite categories.

Lemma 8.4 *If* $F \colon \mathsf{K} \to \mathsf{K}$ *is an internal endofunctor on a suitable internal category, with canonical bifree algebra* $FA \xrightarrow{a} A$*, then the canonical bifree algebra for* $F^{op} \colon \mathsf{K}^{op} \to \mathsf{K}^{op}$ *is* $F^{op}A \xrightarrow{a^{-1}} A$.

PROOF. By the construction of bifree algebras in Section 7 and the definition of the suitability structure on K^{op}, see Proposition 7.2. □

Lemma 8.5 *Suppose that* $G \colon \mathsf{L} \times \mathsf{K} \to \mathsf{K}$ *is an internal functor where* K *is suitable. Write* $G^{op} \colon \mathsf{L}^{op} \times \mathsf{K}^{op} \to \mathsf{K}^{op}$ *for the evident opposite functor. Then* $(G^\dagger)^{op} = (G^{op})^\dagger \colon \mathsf{L}^{op} \to \mathsf{K}^{op}$.

PROOF. For any object B of L, let $a_B \colon G(B, A_B) \longrightarrow A_B$ be the canonical bifree algebra of $G(B, -) \colon \mathsf{K} \to \mathsf{K}$. Thus $G^\dagger \colon \mathsf{L} \to \mathsf{K}$ is the unique functor with $G^\dagger(B) = A_B$ such that a_B gives the components of a natural transformation from $G((-), A_{(-)}) \colon \mathsf{L} \to \mathsf{K}$ to $G^\dagger$. By Lemma 8.4, the canonical bifree algebra of $G^{op}(B, -) \colon \mathsf{K}^{op} \to \mathsf{K}^{op}$ is $a_B^{-1} \colon G^{op}(B, A_B) \longrightarrow A_B$ in K^{op}. Thus $(G^{op})^\dagger \colon \mathsf{L}^{op} \to \mathsf{K}^{op}$ is the unique functor with $(G^{op})^\dagger(B) = A_B$ such that $a_B^{-1}$is a natural transformation from $G^{op}((-), A_{(-)}) \colon \mathsf{K}^{op} \to \mathsf{K}^{op}$

to $(G^{op})^\dagger$. This property is also satisied by $(G^\dagger)^{op} \colon \mathsf{L}^{op} \to \mathsf{K}^{op}$. Thus indeed $(G^\dagger)^{op} = (G^{op})^\dagger$. $\Box$

To solve recursive domain equations involving functors of mixed variance, it is convenient to replace such functors with derived covariant functors on internal categories of the form $\mathsf{K}^{op} \times \mathsf{K}$. For notational convenience, we shall write $\widehat{\mathsf{K}}$ as an abbreviation for $\mathsf{K}^{op} \times \mathsf{K}$. If K is suitable category then, by Proposition 7.2, so is $\widehat{\mathsf{K}}$. Given an internal functor $F \colon \mathsf{K} \to \mathsf{L}$ we write $\widehat{F}$ for the internal functor $(F^{op} \times F) \colon \widehat{\mathsf{K}} \to \widehat{\mathsf{L}}$. If F is a suitable functor then so is $\widehat{F}$.

We shall be interested in internal functors $F \colon \widehat{\mathsf{K}} \times \cdots \times \widehat{\mathsf{K}} \to \widehat{\mathsf{K}}$ that satisfy an important symmetry condition, cf. [3]. Write $\S \colon (\widehat{\mathsf{K}})^{op} \to \widehat{\mathsf{K}}$ for the internal functor defined by

$$\S(A, B) \;=\; (B, A) \qquad\qquad \S(f, g) \;=\; (g, f). \tag{36}$$

Lemma 8.6 *The functor $\S \colon \widehat{\mathsf{K}}^{op} \to \widehat{\mathsf{K}}$ is suitable.*

PROOF. Immediate from the definitions of the suitability structure on opposite and product categories. $\Box$

Definition 8.7 (Symmetric functor) An internal functor $F \colon \widehat{\mathsf{K}}^k \to \widehat{\mathsf{K}}$ is said to be *symmetric* if the diagram below commutes.

$$\begin{array}{ccc} (\widehat{\mathsf{K}}^{op})^k & \xrightarrow{F^{op}} & \widehat{\mathsf{K}}^{op} \\ {\scriptstyle \S^k}\downarrow & & \downarrow{\scriptstyle \S} \\ \widehat{\mathsf{K}}^k & \xrightarrow[F]{} & \widehat{\mathsf{K}} \end{array}$$

Note that $\S \circ \S$ is the identity functor of $\widehat{\mathsf{K}}$. Thus, if we write $F = \langle F^-, F^+ \rangle$ for the components of a symmetric functor then each of F^- and F^+ determines the other, e.g. we have $F^- \;=\; \S \circ F^+ \circ \S^k$. Trivially, for any $G \colon \mathsf{K} \to \mathsf{K}$, it holds that $\widehat{G} \colon \widehat{\mathsf{K}} \to \widehat{\mathsf{K}}$ is symmetric.

Suppose that K is suitable and $F \colon \widehat{\mathsf{K}}^k \times \widehat{\mathsf{K}} \to \widehat{\mathsf{K}}$ is a symmetric internal functor. Then $F^\dagger \colon \widehat{\mathsf{K}}^k \to \widehat{\mathsf{K}}$ maps any object $\vec{B}$ of $\widehat{\mathsf{K}}^k$ to the canonical bifree algebra of the internal endofunctor $F(\vec{B}, -) \colon \widehat{\mathsf{K}} \to \widehat{\mathsf{K}}$. Just from the fact that $F^\dagger$ parametrically finds bifree algebras, it follows that $F^\dagger$ is naturally isomorphic to a symmetric functor, see [3, §6.4]. However, better than this, by the properties established above, we have:

Lemma 8.8 *For any symmetric internal functor* $F\colon \widehat{\mathsf{K}}^k \times \widehat{\mathsf{K}} \to \widehat{\mathsf{K}}$*, where* K *is suitable, the bifree-algebra finding functor* $F^\dagger\colon \widehat{\mathsf{K}}^k \to \widehat{\mathsf{K}}$ *is symmetric.*

PROOF. By the symmetry of F, the left-hand diagram below commutes.

$$\begin{array}{ccc} (\widehat{\mathsf{K}}^{op})^k \times \widehat{\mathsf{K}}^{op} & \xrightarrow{F^{op}} & \widehat{\mathsf{K}}^{op} \\ \downarrow{\scriptstyle \S^k \times \S} & & \downarrow{\scriptstyle \S} \\ \widehat{\mathsf{K}}^k \times \widehat{\mathsf{K}} & \xrightarrow[F]{} & \widehat{\mathsf{K}} \end{array} \qquad \begin{array}{ccc} (\widehat{\mathsf{K}}^{op})^k & \xrightarrow{(F^{op})^\dagger} & \widehat{\mathsf{K}}^{op} \\ \downarrow{\scriptstyle \S^k} & & \downarrow{\scriptstyle \S} \\ \widehat{\mathsf{K}}^k & \xrightarrow[F^\dagger]{} & \widehat{\mathsf{K}} \end{array}$$

Also $\S\colon \widehat{\mathsf{K}}^{op} \to \widehat{\mathsf{K}}$ is a suitable functor, so the right-hand diagram commutes, by Lemma 8.3. Moreover, by Lemma 8.5, $(F^{op})^\dagger = (F^\dagger)^{op}$. Thus the right-hand diagram expresses the symmetry of $F^\dagger$. $\square$

9 Suitability of pP

We complete the proof of Theorem 1 by establishing the result below.

Proposition 9.1 *If Axiom* **1** *holds then the internal category* pP *is suitable.*

The entire section is devoted to the proof of this proposition.

In this section, for the first time in the proof of Theorem 1, we invoke Axiom **1**. As we use it extensively, we **assume Axiom 1** for the entirety of the section.

The lemma below, which gives a surprisingly useful interpolation condition for establishing well-completeness, is taken from [40].

Lemma 9.2 *An object* X *of* $\mathbf{C}$ *is well-complete if*

$$\mathbf{C} \models \exists p\colon \Sigma.\ (X \textit{ inhabited} \to p) \wedge (p \to X \textit{ well-complete}),$$

where "X *inhabited" means* $\exists x\colon X.\ \top$*.*

PROOF. Suppose that X satisfies the condition. We use Proposition 2.5(1) to show that X is well-complete. Reasoning in $\mathbf{C}$, we have $p\colon \Sigma$ such that X inhabited $\to p$ and $p \to X$ well-complete. Take any functions $q\colon \mathbf{F} \to \Sigma$ and $f\colon (\mathbf{I} \restriction q \circ \iota) \to X$. We must show that there exists a unique $f'\colon (\mathbf{F} \restriction q) \to X$ satisfying $f'(\iota(i)) = f(i)$ for all $i\colon (\mathbf{I} \restriction q \circ \iota)$.

Define $h\colon (\mathbf{I} \restriction q \circ \iota) \to \Sigma$ by $h(i) = p$. Then, given any $i\colon (\mathbf{I} \restriction q \circ \iota)$, we have $f(i)\colon X$, so X is inhabited, whence p holds. Therefore $h(i) = \top$, for all $i\colon (\mathbf{I} \restriction q \circ \iota)$. By Axiom **1**, Σ is well-complete, so there exists a unique $h'\colon (\mathbf{F} \restriction q) \to \Sigma$ satisfying $h'(\iota(i)) = h(i)$ for all $i\colon (\mathbf{I} \restriction q \circ \iota)$. Clearly this equation is satisfied by both $h'\colon i' \mapsto p$ and $h'\colon i' \mapsto \top$. Therefore, given $i'\colon (\mathbf{F} \restriction q)$, we have $p = h'(i') = \top$. This shows that $\mathbf{F} \restriction q$ inhabited implies that p holds.

Consider any $i'\colon (\mathbf{F} \restriction q)$. Given i', we have that p holds, so X is well-complete. Define $g_{i'}\colon (\mathbf{F} \restriction q) \to X$ to be the unique function such that $g_{i'}(\iota(i)) = f(i)$. The notation is chosen to emphasise that the definition of $g_{i'}$ depends on the existence of i'.

At last, define $f'\colon (\mathbf{F} \restriction q) \to X$ by $f'(i') = g_{i'}(i')$. We must show that this is the unique function satisfying $f'(\iota(i)) = f(i)$ for all $i\colon (\mathbf{I} \restriction q \circ \iota)$. To see that the equation holds, take any $i\colon (\mathbf{I} \restriction q \circ \iota)$. Then indeed $f'(\iota(i)) = g_{\iota(i)}(\iota(i)) = f(i)$. For uniqueness, suppose that $f''\colon (\mathbf{F} \restriction q) \to X$ satisfies $f''(\iota(i)) = f(i)$, for all $i\colon (\mathbf{I} \restriction q \circ \iota)$. Then, for any $i'\colon (\mathbf{F} \restriction q)$, we have $f'' = g_{i'}$, by the uniqueness of $g_{i'}$. Thus indeed $f''(i') = g_{i'}(i') = f'(i')$. $\square$

By relativizing the above lemma to the slice category $\mathbf{C}/\mathcal{P}_{\mathbf{S}}U$ (where U is the universal object of $\mathbf{C}$), it can be used internally to establish the well-completeness of any $X\colon \mathcal{P}_{\mathbf{S}}U$.

Lemma 9.3 *The morphism* $\bigcup\colon \mathbb{L}(\mathcal{P}_{\mathbf{S}}U) \longrightarrow \mathcal{P}_{\mathbf{S}}U$ *restricts to a morphism* $\bigcup\colon \mathbb{L}|\mathsf{pP}| \longrightarrow |\mathsf{pP}|$*, giving a pointed structure* $(|\mathsf{pP}|, \bigcup)$*.*

PROOF. We first show that $\bigcup$ restricts to give a morphism $\mathbb{L}|\mathsf{pP}| \longrightarrow |\mathsf{pP}|$, i.e. if e is a Σ-subterminal subset of $\mathcal{P}_{\mathbf{S}}U$ whose only element, if it exists, is a predomain then $\bigcup e$ is a predomain. Suppose then that e is Σ-subterminal and $X \in e$ implies X is a predomain.

Trivially $\bigcup e$ is small. We use Lemma 9.2 to show that $\bigcup e$ is well-complete, taking p to be the proposition "e inhabited", which is in Σ because e is Σ-subterminal. The condition "$(\bigcup e)$ inhabited $\to p$" holds trivially. To show that "$p \to \bigcup e$ well-complete" holds, suppose e is inhabited. Then, because e is subterminal, $e = \{X\}$ for some predomain X. Thus $\bigcup e = X$ which is a predomain, hence well-complete.

It remains to show that $\bigcup : \mathbb{L}|\mathsf{pP}| \longrightarrow |\mathsf{pP}|$ satisfies equations (6) and (7) for being an Eilenberg-Moore algebra. The unit law (6) is trivial. For the multiplication law, (7), we must show that $\bigcup(\bigcup E) = \bigcup(\{\bigcup(e) \mid e \in E\})$ for any $E\colon L^2|\mathsf{pP}|$. To show $\bigcup(\bigcup E) \subseteq \bigcup(\{\bigcup(e) \mid e \in E\})$, suppose $x\colon U$ is such that $x \in \bigcup(\bigcup E)$. Then there exist X, e such that $x \in X \in e \in E$. Thus $x \in \bigcup(e)$,

whence $x \in \bigcup(\{\bigcup(e) \mid e \in E\})$. Conversely, if $x \in \bigcup(\{\bigcup(e) \mid e \in E\})$, then there exists $e \in E$ with $x \in \bigcup(e)$, whence there exists X such that $x \in X \in e$. Thus indeed $x \in \bigcup(\bigcup E)$. $\square$

Lemma 9.4 *For $A, B : |\mathsf{pP}|$, it holds that $\mathsf{pP}(A, B)$ is pointed and complete.*

PROOF. We have $\mathsf{pP}(A, B) = A \rightharpoonup B \cong \mathbb{L}B^A$. The latter object is pointed because B is and pointed objects are closed under internal powers (as a special case of internal products). As Axiom **1** holds, we have that $\mathbb{L}B^A$ is complete by the cartesian closure of P and its closure under lifting L. $\square$

It is convenient to have an explicit description of the pointed structure on $A \rightharpoonup B$. This is given by $\beta_{AB} : \mathbb{L}(A \rightharpoonup B) \to (A \rightharpoonup B)$ defined by

$$\beta_{AB}(e)(x) = y \text{ iff there exists } f \in e \text{ with } f(x) = y\,.$$

N.b. here we use $=$ to mean strict equality, see Section 2.

Lemma 9.5 *The composition map $\mathsf{pP}(B, C) \times \mathsf{pP}(A, B) \longrightarrow \mathsf{pP}(A, C)$ is bistrict, for all $A, B, C : |\mathsf{pP}|$.*

PROOF. To show strictness in the first argument, take any $e : \mathbb{L}(B \rightharpoonup C)$ and $f : A \rightharpoonup B$. We must show that $\beta_{BC}(e) \circ f = \beta_{AC}(\{g \circ f \mid g \in e\})$. But, for any $x : A$, we indeed have:

$$\begin{aligned}(\beta_{BC}(e) \circ f)(x) = z &\text{ iff there exists } g \in e \text{ such that } g(f(x)) = z \\ &\text{iff } \beta_{AC}(\{g \circ f \mid g \in e\})(x) = z\,.\end{aligned}$$

For strictness in the second argument, take any $g : B \rightharpoonup C$ and $e : \mathbb{L}(A \rightharpoonup B)$. A similar argument, shows that $g \circ \beta_{AB}(e) = \beta_{AC}(\{g \circ f \mid f \in e\})$ as required. $\square$

Lemma 9.6 $\{\mathsf{id}_A\}_{A : |\mathsf{pP}|}$ *is a strict family.*

PROOF. Because Axiom **1** holds, $(|\mathsf{pP}|, \bigcup)$ is pointed, by Lemma 9.3. We must show equation (8), i.e. that $\mathsf{id}_{\bigcup e} = \beta_{\bigcup e \bigcup e}(\{\mathsf{id}_A \mid A \in e\})$, for all $e : \mathbb{L}|\mathsf{pP}|$. Take any $e : \mathbb{L}|\mathsf{pP}|$ and $x \in \bigcup e$. Then $e = \{A\}$ for some A with $x \in A$. So $\beta_{\bigcup e \bigcup e}(\{\mathsf{id}_A \mid A \in e\})(x) = \mathsf{id}_A(x)$ as required. $\square$

It remains to establish that standard **I**-bichains in pP have specified bilimits. For this, it is convenient to relate pP to the internal category P defined in Section 3. Recall that Axiom **1** implies that P carries a monad $(L, \{\cdot\}, \bigcup)$

internalizing $\mathbb{L}$. Moreover, internally in $\mathbf{C}$, the internal Kleisli category P_L is isomorphic to pP. Thus there is an internal full and faithful comparison functor $K : \mathsf{pP} \to \mathsf{P}^L$, where P^L is the internal Eilenberg-Moore category for the monad. As for any monadic functor, the internal forgetful functor $U : \mathsf{P}^L \to \mathsf{P}$ creates limits. Thus P^L is internally small-complete, because P is.

To obtain bilimits for $\mathbf{I}$-bichains in pP, it suffices, by the limit-colimit coincidence, to construct limits for the diagrams $(x_{ij})_{i \sqsupseteq j}$ derived from $\mathbf{I}$-bichains. We show that arbitrary diagrams of shape $(\mathbf{I}, \sqsupseteq)$ in pP have limits.

Lemma 9.7 *The comparison functor $K : \mathsf{pP} \to \mathsf{P}^L$ creates (up to isomorphism) limits for diagrams of shape $(\mathbf{I}, \sqsupseteq)$.*

PROOF. Suppose we have a diagram $(x_{ij} \colon A_i \rightharpoonup A_j)_{i \sqsupseteq j}$ in pP. Its image under $K : \mathsf{pP} \to \mathsf{P}^L$ is a diagram $(\overline{x_{ij}} : LA_i \to LA_j)_{i \sqsupseteq j}$, whose limit in P^L has underlying set:

$$B \;=\; \{\widetilde{a}_{(-)} \colon \prod_{i \colon \mathbf{I}} LA_i \mid \forall\, i, j \colon \mathbf{I}.\;\; i \sqsupseteq j \text{ implies } \widetilde{a}_j = \overline{x_{ij}}(\widetilde{a}_i)\}\,.$$

By Axiom **1**, the dominance Σ is complete. Thus we can define a subobject

$$B' \;=\; \{\widetilde{a}_{(-)} : B \;\mid\; \bigsqcup_i^{\Sigma} (\widetilde{a}_i{\downarrow})\},$$

where we write $\widetilde{a}_i{\downarrow}$ for the Σ-property $\exists a \colon A_i.\; a \in \widetilde{a}_i$, as in Proposition 4.1.

Define $f : B \to LB'$ and $g : LB' \to B$ by:

$$f(\widetilde{a}_{(-)}) \;=\; \{\widetilde{a}_{(-)} \;\mid\; \bigsqcup_i^{\Sigma} (\widetilde{a}_i{\downarrow})\} \qquad (g(\widetilde{b}))_i \;=\; \bigcup \{\widetilde{a}_i \;\mid\; \widetilde{a}_{(-)} \in \widetilde{b}\}$$

We show that f and g are mutual inverses.

For the identity $g \circ f = \mathsf{id}_B$, we have $(g(f(\widetilde{a}_{(-)})))_i \;=\; \bigcup \{\widetilde{a}_i \;\mid\; \bigsqcup_j^{\Sigma} (\widetilde{a}_j{\downarrow})\}$. So we must show that $\widetilde{a}_i = \bigcup \{\widetilde{a}_i \;\mid\; \bigsqcup_j^{\Sigma} (\widetilde{a}_j{\downarrow})\}$. We establish subset inclusions in each direction. Clearly $\widetilde{a}_i \supseteq \bigcup \{\widetilde{a}_i \;\mid\; \bigsqcup_j^{\Sigma} (\widetilde{a}_j{\downarrow})\}$. For the converse, it suffices to show that, for all $i \colon \mathbf{I}$, if $\widetilde{a}_i{\downarrow}$ then $\bigsqcup_j^{\Sigma} (\widetilde{a}_j{\downarrow})$. By the definition of B, we have that $i \sqsubseteq j$ implies $\widetilde{a}_i = \overline{x_{ji}}(\widetilde{a}_j)$, where $\overline{x_{ji}}$ is strict. Thus, by Proposition 4.1, if $i \sqsubseteq j$ and $\widetilde{a}_i{\downarrow}$ then $\widetilde{a}_j{\downarrow}$; i.e. $i \sqsubseteq j$ implies $\widetilde{a}_i{\downarrow} = \;\widetilde{a}_i{\downarrow} \wedge \widetilde{a}_j{\downarrow}$ in Σ. Now take any $i \colon \mathbf{I}$. Then

$$\begin{aligned}
\widetilde{a}_i{\downarrow} \;&=\; \bigsqcup_j^{\Sigma} (\widetilde{a}_{\min(i,j)}{\downarrow}) && \text{by (20)} \\
&=\; \bigsqcup_j^{\Sigma} (\widetilde{a}_{\min(i,j)}{\downarrow} \;\wedge \widetilde{a}_j{\downarrow}) && \text{as } \min(i,j) \sqsubseteq j
\end{aligned}$$

$$
\begin{aligned}
&= \bigsqcup_j^{\Sigma} (\widetilde{a}_{\min(i,j)}\!\downarrow) \;\wedge\; \bigsqcup_j^{\Sigma} (\widetilde{a}_j\!\downarrow) && \text{by (23) and (24)}\\
&= \widetilde{a}_i\!\downarrow \;\wedge \bigsqcup_j^{\Sigma} (\widetilde{a}_j\!\downarrow) && \text{by (20).}
\end{aligned}
$$

But this equality states that $\widetilde{a}_i\!\downarrow$ implies $\bigsqcup_j^{\Sigma} (\widetilde{a}_j\!\downarrow)$, as required.

To show that $f \circ g = \mathsf{id}_{LB'}$, for any $\widetilde{b}\colon LB'$, we have that

$$
f(g(\widetilde{b})) \;=\; \{\, \{\bigcup \{\widetilde{a}_i \mid \widetilde{a}_{(-)} \in \widetilde{b}\}\}_{i:\mathbf{I}} \;\mid\; \bigsqcup_i^{\Sigma} \,(\,(\bigcup \{\widetilde{a}_i \mid \widetilde{a}_{(-)} \in \widetilde{b}\})\!\downarrow)\,\}. \qquad (37)
$$

Given any $\widetilde{b}\colon LB'$, suppose that $\widetilde{a}_{(-)} \in \widetilde{b}$. Then $\bigcup\{\widetilde{a}_i \mid \widetilde{a}_{(-)} \in \widetilde{b}\} = \widetilde{a}_i$ and also $\bigsqcup_i^{\Sigma} (\widetilde{a}_i\!\downarrow)$ because $\widetilde{a}_{(-)}\colon B'$. So, by (37), it is clear that $\widetilde{b} \subseteq f(g(\widetilde{b}))$.

For the converse inclusion, suppose $\widetilde{a}_{(-)} \in f(g(\widetilde{b}))$. Then, by (37), we have: $\widetilde{a}_i \;=\; \bigcup\{\widetilde{a}'_i \mid \widetilde{a}'_{(-)} \in \widetilde{b}\}$. It follows that, $\widetilde{a}_i\!\downarrow$ implies $\widetilde{b}\!\downarrow$, for all $i\colon\mathbf{I}$. In other words, $\widetilde{a}_i\!\downarrow = \widetilde{a}_i\!\downarrow \wedge \widetilde{b}\!\downarrow$ in Σ. It also follows that $\widetilde{a}'_{(-)} \in \widetilde{b}$ implies $\widetilde{a}'_i = \widetilde{a}_i$ for all $i\colon\mathbf{I}$. Thus $\widetilde{b}\!\downarrow$ implies $\widetilde{a}_{(-)} \in \widetilde{b}$. But indeed $\widetilde{b}\!\downarrow$ because:

$$
\begin{aligned}
\widetilde{b}\!\downarrow \;&=\; \widetilde{b}\!\downarrow \;\wedge \bigsqcup_i^{\Sigma} (\widetilde{a}_i\!\downarrow) && \text{because } \widetilde{a}_{(-)}\colon B'\\
&=\; \bigsqcup_i^{\Sigma} (\widetilde{b}\!\downarrow \;\wedge\; \widetilde{a}_i\!\downarrow) && \text{by (24)}\\
&=\; \bigsqcup_i^{\Sigma} (\widetilde{a}_i\!\downarrow) && \text{as } \widetilde{a}_i\!\downarrow = \widetilde{a}_i\!\downarrow \wedge \widetilde{b}\!\downarrow\\
&=\; \top && \text{because } \widetilde{a}_{(-)}\colon B'.
\end{aligned}
$$

So indeed $f(g(\widetilde{b})) \subseteq \widetilde{b}$, and hence $f \circ g = \mathsf{id}_{LB'}$.

Next we show that g is strict, i.e. is a morphism in P^L. The pointed structure on B is $\beta\colon LB \to B$ defined by

$$
\beta(\widetilde{b})_i \;=\; \bigcup\{\widetilde{a}_i \mid \widetilde{a}_{(-)} \in \widetilde{b}\}\,.
$$

Given $E\colon L^2B$, we show that $g(\bigcup E) = \beta(\{g(\widetilde{b}) \mid \widetilde{b} \in E\})$ by:

$$
\begin{aligned}
\beta(\{g(\widetilde{b}) \mid \widetilde{b} \in E\})_i \;&=\; \bigcup\{g(\widetilde{b})_i \mid \widetilde{b} \in E\} && \text{def. of } \beta\\
&=\; \bigcup\{\bigcup\{\widetilde{a}_i \mid \widetilde{a}_{(-)} \in \widetilde{b}\} \mid \widetilde{b} \in E\} && \text{def. of } g\\
&=\; \bigcup\{\widetilde{a}_i \mid \widetilde{a}_{(-)} \in \bigcup E\}\\
&=\; g(\bigcup E)_i && \text{def. of } g.
\end{aligned}
$$

Thus g is indeed strict.

As $U : \mathsf{P}^L \to \mathsf{P}$ is a monadic functor, it reflects isomorphisms. Thus g is an isomorphism in P^L (with f, which is therefore strict, its inverse). It follows

that the cone $(\pi_i \circ g : LB' \to LA_i)_{i:\mathbf{I}}$ is limiting for $(\overline{x_{ij}} : LA_i \to LA_j)_{i \sqsupseteq j}$, in P^L. Because $K : \mathsf{pP} \to \mathsf{P}^L$ is full and faithful, the above cone is the image of a limiting cone $q_{(-)} : \prod_{i:\mathbf{I}} B' \rightharpoonup A_i$ for the diagram $(x_{ij} : A_i \rightharpoonup A_j)_{i \sqsupseteq j}$ in pP. □

Corollary 9.8 *The internal category* pP *has limits for* $(\mathbf{I}, \sqsupseteq)$ *diagrams.*

PROOF. Immediate from Lemma 9.7, because P^L is small-complete. □

It follows from the proof above that there is a morphism in $\mathbf{C}$ mapping $\mathbf{I}$-opchains in pP to their limiting cones. Indeed, this follows by unwinding the proof, which constructs the limit-finding morphism from that for P^L, which is, in turn, obtained from that for P, which exists by the internal version of Proposition 2.5. Moreover, the morphism mapping $\mathbf{I}$-opchains to their limiting cones determines a morphism mapping $\mathbf{I}$-bichains to their bilimits, because $l_{(-)}$ determines $c_{(-)}$ in Proposition 6.2. The lemma below summarises this.

Lemma 9.9 *Every* $\mathbf{I}$*-bichain in* pP *has a bilimit and the operation mapping* $\mathbf{I}$*-bichains to bilimits is given by a morphism in* $\mathbf{C}$*.*

Taken together, Lemmas 9.3–9.9 prove Proposition 9.1, and hence Theorem 1.

10 The language FPC

In this section, we give a brief overview of Plotkin's call-by-value recursively typed λ-calculus, FPC, introduced in [33]. For full details see [3].

We use $X, Y, \ldots$ to range over type variables, and $\sigma, \tau, \ldots$ to range over types, which are given by:

$$\sigma ::= X \mid \sigma + \tau \mid \sigma \times \tau \mid \sigma \to \tau \mid \mu X.\sigma.$$

Here the prefix μX binds X. We use $\Theta, \ldots$ to range over finite sequences of distinct type variables. We write $\Theta \vdash \sigma$ to mean that all free type variables in σ appear in Θ.

We use $x, y, \ldots$ to range over term variables, and $s, t, \ldots$ to range over terms, which are given by:

$$\begin{aligned} t \ ::= {} & x \mid \mathtt{inl}(t) \mid \mathtt{inr}(t) \mid \mathtt{case}(s)\ \mathtt{of}\ x_1.t_1\ \mathtt{or}\ x_2.t_2 \mid (s,t) \mid \mathtt{fst}(t) \mid \mathtt{snd}(t) \mid \\ & \lambda x.\, t \mid s(t) \mid \mathtt{intro}(t) \mid \mathtt{elim}(t). \end{aligned}$$

$$\frac{}{\Gamma,\, x\colon \sigma \vdash x\colon \sigma} \qquad \frac{\Gamma \vdash t\colon \sigma}{\Gamma \vdash \mathtt{inl}(t)\colon \sigma + \tau} \qquad \frac{\Gamma \vdash t\colon \tau}{\Gamma \vdash \mathtt{inr}(t)\colon \sigma + \tau}$$

$$\frac{\Gamma \vdash s\colon \sigma_1 + \sigma_2 \quad \Gamma,\, x_1\colon \sigma_1 \vdash t_1\colon \tau \quad \Gamma,\, x_2\colon \sigma_2 \vdash t_2\colon \tau}{\Gamma \vdash \mathtt{case}(s)\ \mathtt{of}\ x_1.t_1\ \mathtt{or}\ x_2.t_2\colon \tau}$$

$$\frac{\Gamma \vdash s\colon \sigma \quad \Gamma \vdash t\colon \tau}{\Gamma \vdash (s,t)\colon \sigma \times \tau} \qquad \frac{\Gamma \vdash t\colon \sigma \times \tau}{\Gamma \vdash \mathtt{fst}(t)\colon \sigma} \qquad \frac{\Gamma \vdash t\colon \sigma \times \tau}{\Gamma \vdash \mathtt{snd}(t)\colon \tau}$$

$$\frac{\Gamma,\, x\colon \sigma \vdash t\colon \tau}{\Gamma \vdash \lambda x.\, t\colon \sigma \to \tau} \qquad \frac{\Gamma \vdash t\colon \sigma \to \tau \quad \Gamma \vdash s\colon \sigma}{\Gamma \vdash t(s)\colon \tau}$$

$$\frac{\Gamma \vdash t\colon \sigma[\mu X.\sigma/X]}{\Gamma \vdash \mathtt{intro}(t)\colon \mu X.\sigma} \qquad \frac{\Gamma \vdash t\colon \mu X.\sigma}{\Gamma \vdash \mathtt{elim}(t)\colon \sigma[\mu X.\sigma/X]}$$

Fig. 1. Typing rules for FPC.

$$\frac{t \rightsquigarrow v}{\mathtt{inl}(t) \rightsquigarrow \mathtt{inl}(v)} \qquad \frac{t \rightsquigarrow v}{\mathtt{inr}(t) \rightsquigarrow \mathtt{inr}(v)}$$

$$\frac{s \rightsquigarrow \mathtt{inl}(v_1) \quad t_1[v_1/x_1] \rightsquigarrow v}{\mathtt{case}(s)\ \mathtt{of}\ x_1.t_1\ \mathtt{or}\ x_2.t_2 \rightsquigarrow v} \qquad \frac{s \rightsquigarrow \mathtt{inr}(v_2) \quad t_2[v_2/x_2] \rightsquigarrow v}{\mathtt{case}(s)\ \mathtt{of}\ x_1.t_1\ \mathtt{or}\ x_2.t_2 \rightsquigarrow v}$$

$$\frac{t_1 \rightsquigarrow v_1 \quad t_2 \rightsquigarrow v_2}{(t_1,t_2) \rightsquigarrow (v_1,v_2)} \qquad \frac{t \rightsquigarrow (v_1,v_2)}{\mathtt{fst}(t) \rightsquigarrow v_1} \qquad \frac{t \rightsquigarrow (v_1,v_2)}{\mathtt{snd}(t) \rightsquigarrow v_2}$$

$$\frac{}{\lambda x.\, t \rightsquigarrow \lambda x.\, t} \qquad \frac{t \rightsquigarrow \lambda x.\, t' \quad s \rightsquigarrow v' \quad t'[v'/x] \rightsquigarrow v}{t(s) \rightsquigarrow v}$$

$$\frac{t \rightsquigarrow v}{\mathtt{intro}(t) \rightsquigarrow \mathtt{intro}(v)} \qquad \frac{t \rightsquigarrow \mathtt{intro}(v)}{\mathtt{elim}(t) \rightsquigarrow v}$$

Fig. 2. Evaluation rules for FPC.

We use $\Gamma, \ldots$ to range over sequences of the form $x_1 \colon \sigma_1, \ldots, x_k \colon \sigma_k$ with all x_i distinct and all σ_i closed. For closed types σ, we write $\Gamma \vdash t \colon \sigma$ to mean that t is a well-formed term of type σ relative to Γ, where the rules for deriving such typing assertions are given in Fig. 1. We view a term as uniquely determining its typing derivation. In order to achieve this formally, one should properly include type information in the terms $\mathtt{inl}(t)$, $\mathtt{inr}(t)$, $\mathtt{intro}(t)$ and $\lambda x.\, t$, see [3]. We consider such type information as being there implicitly, but to ease clutter we never write it.

To define a call-by-value operational semantics for FPC, we first specify the *values*, closed terms $v, \ldots$ of the form:

$$v \ ::= \ \mathtt{inl}(v) \mid \mathtt{inr}(v) \mid (v_1, v_2) \mid \lambda x.\, t \mid \mathtt{intro}(v).$$

The call-by-value evaluation relation $t \rightsquigarrow v$ between closed terms t and values v is defined in Fig. 2. Importantly, if $\vdash t \colon \sigma$ and $t \rightsquigarrow v$ then $\vdash v \colon \sigma$. We say that a closed term t *converges*, notation $t \Downarrow$, if there exists (a necessarily unique) v such that $t \rightsquigarrow v$.

In Section 14, we shall need to do some simple programming in FPC. To facilitate this, we define various useful datatypes and operations upon them.

Basic datatypes, including a type of natural numbers, are encoded by:

$$\begin{aligned} \mathtt{empty} &= \mu X.\, X \\ \mathtt{unit} &= \mathtt{empty} \to \mathtt{empty} \\ \mathtt{bool} &= \mathtt{unit} + \mathtt{unit} \\ \mathtt{nat} &= \mu X.\, \mathtt{unit} + X\,. \end{aligned}$$

The standard operations on such datatypes are easily defined: a canonical element $*$ of $\mathtt{unit}$; the truth values $\mathtt{tt}$ and $\mathtt{ff}$ in $\mathtt{bool}$ along with an associated $\mathtt{if} \ldots \mathtt{then} \ldots \mathtt{else} \ldots$ construction; numerals $\overline{n} \colon \mathtt{nat}$, for each $n \in \mathbb{N}$, successor and predecessor functions $\mathtt{succ}, \mathtt{pred} \colon \mathtt{nat} \to \mathtt{nat}$, and an equality predicate of type $\mathtt{nat} \times \mathtt{nat} \to \mathtt{bool}$. Crucially, FPC also supports the recursive definition of functions. Given a term $\Gamma,\ f \colon \sigma \to \tau,\ x \colon \sigma \vdash t \colon \tau$, write $\mathtt{rec}\, f = \lambda x.\, t$ for the term $\delta(\mathtt{intro}(\delta))$, where

$$\delta \ = \ \lambda w.\ (\lambda z.\, \lambda x.\, t\,[\, z\,(\mathtt{intro}(z))\,/\,f\,]\,)\,(\mathtt{elim}(w))\,.$$

Then, by giving w the type $\mu X.\, (X \to (\sigma \to \tau))$, one derives that

$$\Gamma \vdash \mathtt{rec}\, f = \lambda x.\, t \colon \sigma \to \tau\,.$$

Moreover, $\mathtt{rec}\, f = \lambda x.\, t$ enjoys the following operational behaviour:

$$\mathtt{rec}\, f = \lambda x.\, t \ \rightsquigarrow \ \lambda x.\, t\,[\,\mathtt{rec}\, f = \lambda x.\, t\,/\,f\,]\,.$$

Thus $\mathtt{rec}\, f = \lambda x.\, t$ indeed implements the recursive definition of functions.

11 The interpretation of FPC

In this Section, we apply Theorem 1 to obtain an interpretation of FPC, in the category **pP**. It is convenient to first define an interpretation in the internal category pP and then to extract from that the interpretation in **pP**. Moreover, although the interpretation in pP could be obtained using Theorem 1 alone, having developed the technology, it is convenient to make direct use of the properties of suitable categories established in Sections 7 and 8.

To interpret FPC in **pP**, we need **pP** to be closed under $+$, so henceforth, for the remainder of the paper, we **assume Axiom 2**.

To define the interpretation in pP, we first interpret types. To apply algebraic compactness it is necessary to interpret open types as internal functors. Moreover, because of the bivariance of $\rightharpoonup$, they must be interpreted as internal functors on the internal category $\mathsf{pP}^{op} \times \mathsf{pP}$, for which we write $\widehat{\mathsf{pP}}$, as in Section 8. The functors will all be symmetric in the sense of Definition 8.7. Indeed, an open type σ is interpreted, relative to any $\Theta = X_1, \ldots, X_k$ such that $\Theta \vdash \sigma$, as a symmetric internal functor,

$$(\![\Theta \vdash \sigma]\!) \colon \widehat{\mathsf{pP}}^{\,k} \to \widehat{\mathsf{pP}}\,.$$

The interpretation is defined by induction on the structure of σ. To give the definition, we write $\vec{A}$ for an object $((A_1^-, A_1^+), \ldots, (A_k^-, A_k^+))$ of $\widehat{\mathsf{pP}}^{\,k}$.

$$\begin{aligned}
(\![X_1, \ldots, X_k \vdash X_i]\!)^+ \vec{A} &= A_i^+ \\
(\![\Theta \vdash \sigma_1 + \sigma_2]\!)^+ \vec{A} &= (\![\Theta \vdash \sigma_1]\!)^+ \vec{A} + (\![\Theta \vdash \sigma_2]\!)^+ \vec{A} \\
(\![\Theta \vdash \sigma_1 \times \sigma_2]\!)^+ \vec{A} &= (\![\Theta \vdash \sigma_1]\!)^+ \vec{A} \times (\![\Theta \vdash \sigma_2]\!)^+ \vec{A} \\
(\![\Theta \vdash \sigma_1 \to \sigma_2]\!)^+ \vec{A} &= (\![\Theta \vdash \sigma_1]\!)^- \vec{A} \rightharpoonup (\![\Theta \vdash \sigma_2]\!)^+ \vec{A} \\
(\![\Theta \vdash \mu X.\sigma']\!) &= (\![\Theta, X \vdash \sigma']\!)^\dagger\,,
\end{aligned}$$

using the internal functors on pP identified in Section 3. Note that, where the above clauses only define the $(\![\Theta \vdash \sigma]\!)^+$ components, the $(\![\Theta \vdash \sigma]\!)^-$ components are determined by symmetry. Also, where the definition is specified on objects, it is extended to morphisms in the obvious way, using the action of the internal functors appearing in the definition. For non-recursive types σ, the symmetry of $(\![\Theta \vdash \sigma]\!)$ is immediate by construction. For recursive types, it holds by Lemma 8.8.

The interpretation of types satisfies a substitution lemma, cf. [3, Lemma 8.4.4]. However, because of the direct interpretation of recursive types using $(\cdot)^\dagger$, we can strengthen the isomorphism of *loc. cit.* to an equality.

Lemma 11.1 *For open types* $\Theta \vdash \tau_1\,,\ \ldots\ ,\ \Theta \vdash \tau_k$ *and* $X_1, \ldots, X_k \vdash \sigma$*,*

$$(\![\, \Theta \vdash \sigma[\vec{\tau}/\vec{X}]\,]\!) \;=\; (\![\, \vec{X} \vdash \sigma\,]\!) \circ \langle (\![\Theta \vdash \tau_1]\!), \ldots, (\![\Theta \vdash \tau_k]\!)\rangle\,.$$

PROOF. A straightforward induction on types, using Proposition 3.5 in the case for recursive types. □

For closed types, the functor $(\![\vdash \sigma]\!)\colon \mathsf{1} \to \widehat{\mathsf{pP}}$, where $\mathsf{1}$ is the terminal internal category, corresponds, by symmetry, to an object in $\widehat{\mathsf{pP}}$ of the form (A, A). We write $(\![\sigma]\!)$ for the corresponding object A of pP.

For a closed recursive type, $\mu X.\sigma$, we have that $(\![\vdash \mu X.\sigma]\!)$, i.e. the object $((\![\mu X.\sigma]\!), (\![\mu X.\sigma]\!))$, carries the canonical bifree algebra structure for the symmetric functor $(\![X \vdash \sigma]\!)\colon \widehat{\mathsf{pP}} \to \widehat{\mathsf{pP}}$. The bifree algebra is an isomorphism,

$$(\epsilon_{\mu X.\sigma}, \iota_{\mu X.\sigma})\colon (\![X \vdash \sigma]\!)\,((\![\mu X.\sigma]\!), (\![\mu X.\sigma]\!)) \longrightarrow ((\![\mu X.\sigma]\!), (\![\mu X.\sigma]\!))\,,$$

in $\widehat{\mathsf{pP}}$. By Lemma 11.1, this gives

$$(\epsilon_{\mu X.\sigma}, \iota_{\mu X.\sigma})\colon (\![\vdash \sigma[\mu X.\sigma/X]]\!) \longrightarrow ((\![\mu X.\sigma]\!), (\![\mu X.\sigma]\!))\,,$$

which unpacks to give isomorphisms in pP

$$\iota_{\mu X.\sigma}\colon (\![\sigma[\mu X.\sigma/X]]\!) \longrightarrow (\![\mu X.\sigma]\!) \tag{38}$$
$$\epsilon_{\mu X.\sigma}\colon (\![\mu X.\sigma]\!) \longrightarrow (\![\sigma[\mu X.\sigma/X]]\!)\,. \tag{39}$$

Moreover, as $(\![X \vdash \sigma]\!)$ is symmetric and § is a suitable functor, it follows from Lemmas 8.2 and 8.4 that $\iota_{\mu X.\sigma} = {\epsilon_{\mu X.\sigma}}^{-1}$.

To interpret the terms of FPC in pP, a context $\Gamma = x_1\colon \sigma_1, \ldots, x_k\colon \sigma_k$ is interpreted as the object $(\![\Gamma]\!) = (\![\sigma_1]\!) \times \cdots \times (\![\sigma_k]\!)$ of pP. A term $\Gamma \vdash t\colon \sigma$ is interpreted as a morphism $(\![t]\!)_\Gamma$ from $(\![\Gamma]\!)$ to $(\![\sigma]\!)$ in pP, i.e. as a point $(\![t]\!)_\Gamma\colon \mathbf{1} \longrightarrow \mathsf{pP}((\![\Gamma]\!), (\![\sigma]\!))$ in $\mathbf{C}$, or equivalently as an internal partial function $(\![t]\!)_\Gamma\colon (\![\Gamma]\!) \rightharpoonup (\![\sigma]\!)$. As is standard, the definition of $(\![t]\!)_\Gamma$ is by induction on the structure of t. To give the definition, we use evident notation for application of partial functions, and we use Kleene equality $\simeq$, see Section 2. Moreover, we extend the pairing and injection functions to act strictly on possibly undefined expressions; i.e. the result is defined (if and) only if all arguments are defined. Then, internally in $\mathbf{C}$, we understand $(\![t]\!)_\Gamma\colon (\![\Gamma]\!) \rightharpoonup (\![\sigma]\!)$ to be the

least-defined[3] partial function satisfying, for $\vec{d}$: $(\!|\Gamma|\!)$:

$$\begin{array}{rcll}
(\!|x_i|\!)_\Gamma(d_1,\ldots,d_k) & = & d_i & \\
(\!|\mathtt{inl}(t)|\!)_\Gamma(\vec{d}) & \simeq & \mathrm{inl}((\!|t|\!)_\Gamma(\vec{d})) & \\
(\!|\mathtt{inr}(t)|\!)_\Gamma(\vec{d}) & \simeq & \mathrm{inr}((\!|t|\!)_\Gamma(\vec{d})) & \\
(\!|\mathtt{case}(s)\ \mathtt{of}\ x_1.t_1\ \mathtt{or}\ x_2.t_2|\!)_\Gamma(\vec{d}) & \simeq & [\![t_1]\!]_{\Gamma,x_1:\,\sigma_1}(\vec{d},c) & \text{if } [\![s]\!]_\Gamma(\vec{d}) = \mathrm{inl}(c) \\
(\!|\mathtt{case}(s)\ \mathtt{of}\ x_1.t_1\ \mathtt{or}\ x_2.t_2|\!)_\Gamma(\vec{d}) & \simeq & [\![t_2]\!]_{\Gamma,x_2:\,\sigma_2}(\vec{d},c) & \text{if } [\![s]\!]_\Gamma(\vec{d}) = \mathrm{inr}(c) \\
(\!|(s,t)|\!)_\Gamma(\vec{d}) & \simeq & ((\!|s|\!)_\Gamma(\vec{d}),(\!|t|\!)_\Gamma(\vec{d})) & \\
(\!|\mathtt{fst}(t)|\!)_\Gamma(\vec{d}) & \simeq & \pi_1((\!|t|\!)_\Gamma(\vec{d})) & \\
(\!|\mathtt{snd}(t)|\!)_\Gamma(\vec{d}) & \simeq & \pi_2((\!|t|\!)_\Gamma(\vec{d})) & \\
(\!|\lambda x.\,t|\!)_\Gamma(\vec{d}) & = & (c \mapsto (\!|t|\!)_{\Gamma,x:\,\sigma}(\vec{d},c)) & \\
(\!|s(t)|\!)_\Gamma(\vec{d}) & \simeq & (\!|s|\!)_\Gamma(\vec{d})((\!|t|\!)_\Gamma(\vec{d})) & \\
(\!|\mathtt{intro}(t)|\!)_\Gamma(\vec{d}) & \simeq & \iota_{\mu X.\sigma}((\!|t|\!)_\Gamma(\vec{d})) & \\
(\!|\mathtt{elim}(t)|\!)_\Gamma(\vec{d}) & \simeq & \epsilon_{\mu X.\sigma}((\!|t|\!)_\Gamma(\vec{d})). &
\end{array}$$

Here we are assuming that the types of subterms are as in the typing rules of Fig. 1. Note that the above definition defines $(\!|t|\!)_\Gamma$ by an *external* induction on the structure of t as the *internal* partial function determined by the above *internal* (Kleene) equalities. When Γ is empty (i.e. t is closed), we simply write $(\!|t|\!)$ for $(\!|t|\!)_\Gamma$.

Having now obtained the *internal* interpretation of FPC, in the internal category pP, we extract an *external* "real world" interpretation in the category $\mathbf{pP}$. A closed type σ is interpreted as an object $[\![\sigma]\!]$ of $\mathbf{pP}$, by defining $[\![\sigma]\!]$ as the pullback below.

$$\begin{array}{ccccc}
[\![\sigma]\!] & \longrightarrow & & & \ni_U \\
\downarrow & \lrcorner & & & \downarrow{\scriptstyle \gamma_U} \\
\mathbf{1} & \xrightarrow[(\!|\sigma|\!)]{} & |\mathsf{pP}| & \rightarrowtail & \mathcal{P}_{\mathbf{S}}U\,,
\end{array} \tag{40}$$

where γ_U is as in (1) from Section 2. The object $[\![\sigma]\!]$ is indeed a predomain by the definition of $|\mathsf{pP}|$ as a subobject of $\mathcal{P}_{\mathbf{S}}U$. Similarly, a context Γ is interpreted as an object $[\![\Gamma]\!]$, by replacing σ with Γ in the diagram above. We interpret a term $\Gamma \vdash t\colon \sigma$ as a morphism $[\![t]\!]_\Gamma\colon [\![\Gamma]\!] \longrightarrow [\![\sigma]\!]$ in $\mathbf{pP}$, by trans-

[3] The qualification "least-defined" is inserted to ensure that $(\!|\mathtt{case}(s)\ \mathtt{of}\ x_1.t_1\ \mathtt{or}\ x_2.t_2|\!)_\Gamma(\vec{d})$ is defined only if $[\![s]\!]_\Gamma(\vec{d}) = \mathrm{inl}(c)$ or $[\![s]\!]_\Gamma(\vec{d}) = \mathrm{inr}(c)$. This could alternatively be achieved by using a more complex single clause to define $(\!|\mathtt{case}(s)\ \mathtt{of}\ x_1.t_1\ \mathtt{or}\ x_2.t_2|\!)_\Gamma(\vec{d})$.

posing $(\!|t|\!)_\Gamma : \mathsf{pP}((\!|\Gamma|\!), (\!|\sigma|\!))$, i.e. $(\!|t|\!)_\Gamma : (\!|\Gamma|\!) \rightharpoonup (\!|\sigma|\!)$, in the evident way. When Γ is empty, we write simply $[\![t]\!] : \mathbf{1} \longrightarrow [\![\sigma]\!]$.

We remark that there is another way of viewing the above external interpretation of FPC. As for any internal category, the internal structure of pP, used to define the internal interpretation of FPC, corresponds to external structure on the fibration $\mathbf{Ext}(\mathsf{pP}) \to \mathbf{C}$, see Section 3. By the definition of externalization[17, §7.3], the fibre $\mathbf{Ext}(\mathsf{pP})_\mathbf{1}$ over the terminal object $\mathbf{1}$ is given by a category whose objects are points $A \colon \mathbf{1} \longrightarrow |\mathsf{pP}|$. Then, by Proposition 3.2(2), there is an equivalence of categories $I \colon \mathbf{Ext}(\mathsf{pP})_\mathbf{1} \to \mathbf{pP}$ (essentially defined using the pullback (40) above) and $J \colon \mathbf{pP} \to \mathbf{Ext}(\mathsf{pP})_\mathbf{1}$. The various internal functors on pP, e.g. those given in (3)–(5), directly correspond to functors on $\mathbf{Ext}(\mathsf{pP})_\mathbf{1}$ and hence, via I and J, determine functors:

$$\begin{aligned} \mathbf{pP} \times \mathbf{pP} &\xrightarrow{\times'} \mathbf{pP} \\ \mathbf{pP}^{op} \times \mathbf{pP} &\xrightarrow{\rightharpoonup'} \mathbf{pP} \\ \mathbf{pP} \times \mathbf{pP} &\xrightarrow{+'} \mathbf{pP}\,. \end{aligned}$$

Using these functors, one can inductively define an external interpretation of types

$$[\![\Theta \vdash \sigma]\!] \colon (\mathbf{pP}^{op} \times \mathbf{pP})^k \to \mathbf{pP}^{op} \times \mathbf{pP},$$

using clauses analogous to those of the internal definition. Importantly, in the case for recursive types

$$[\![\Theta \vdash \mu X.\sigma']\!] \;=\; [\![\Theta, X \vdash \sigma']\!]^{\dagger'},$$

the operation $(\cdot)^{\dagger'}$, defined by transporting $(\cdot)^\dagger$ on pP along I and J, finds external bifree algebras for each functor $X \mapsto [\![\Theta, X \vdash \sigma']\!](\vec{A}, X)$ (this follows by unwinding the external meaning of Proposition 3.4). In an exactly analogous way, one can also define an interpretation in $\mathbf{pP}$ of FPC terms t, by induction on the structure of t, using the above interpretation of types. Because we have used the internal structure of pP to determine the external structure on $\mathbf{pP}$ applied in these definitions, the above inductive method of interpreting terms in $\mathbf{pP}$ gives rise to the same interpretation $[\![t]\!]_\Gamma \colon [\![\Gamma]\!] \longrightarrow [\![\sigma]\!]$ as defined above. Thus we have shown that $[\![t]\!]_\Gamma \colon [\![\Gamma]\!] \longrightarrow [\![\sigma]\!]$ can be given a direct inductive definition in terms of external structure on $\mathbf{pP}$, as long as the external structure on $\mathbf{pP}$ is determined via I and J from the internal structure of pP. This observation will be important in [44].

The external interpretation of FPC we have given is not as pleasant as one

might like. For example, the interpretation of closed types satisfies,

$$\begin{aligned} [\![\sigma]\!] +' [\![\tau]\!] &= [\![\sigma + \tau]\!] \cong [\![\sigma]\!] + [\![\tau]\!] \\ [\![\sigma]\!] \times' [\![\tau]\!] &= [\![\sigma \times \tau]\!] \cong [\![\sigma]\!] \times [\![\tau]\!] \\ [\![\sigma]\!] \multimap' [\![\tau]\!] &= [\![\sigma \to \tau]\!] \cong [\![\sigma]\!] \multimap [\![\tau]\!], \end{aligned}$$

where the operations on the right are the original specified structure on **pP**. Because the operations $+', \times', \multimap'$ are obtained by externalizing the internal interpretation, there is no reason at all for the isomorphisms to be equalities. Similarly, the interpretation of closed recursive types using $(\cdot)^{\dagger'}$, need not coincide with a given specified external construction of bifree algebras in **pP**. In some ways, it would be more natural to use a different external interpretation under which the isomorphisms and equalities above are swapped (it is not hard to define such an interpretation). For us, the benefit of working with the external interpretation given above is that, by its very definition, it is directly related to the internal interpretation. In a follow-up paper [44], a general coherence theorem is proved, which shows that the properties of interpretations of FPC are anyway independent of such isomorphic choices.

The major remaining goal in this paper is to study a fundamental relationship between denotational and operational semantics. For a closed term $t\colon \sigma$, we write $[\![t]\!]\downarrow$ to mean that the partial map $[\![t]\!]\colon \mathbf{1} \longrightarrow [\![\sigma]\!]$ is total.

Definition 11.2 (Computational adequacy) We say that the interpretation of FPC in **pP** is *computationally adequate* if, for all closed terms $t\colon \sigma$, it holds that $t\Downarrow$ if and only if $[\![t]\!]\downarrow$.

Computational adequacy is equivalent to the soundness of denotational equality in **pP** relative to operational equivalence between terms of FPC.

As our second main result, Theorem 2 below, we shall establish necessary and sufficient conditions for computational adequacy to hold. In order to achieve this, it is useful to first consider a notion computational adequacy formulated within the internal logic of **C**.

12 Internal computational adequacy

Assuming Axiom **2**, we have interpreted FPC in the internal category pP. The main result of this section, Proposition 12.2, is that computational adequacy always holds for this interpretation, when expressed internally in **C**. The proof of this result is rather long. The main proof structure is given in this section, but the verification of several details is left to Section 13.

First we have to formulate the internal notion of computational adeqacy. As with Definition 11.2 above, this will relate the denotational and operational notions of convergence. However, the formulation will use the internal logic of $\mathbf{C}$. On the denotational side we naturally use the interpretation of FPC in the internal category pP. On the operational side, it is necessary to formalize the operational semantics of FPC in the internal logic. To facilitate this, we use a Gödel numbering of the syntax of FPC and its operational semantics. The formalization of such a Gödel numbering in the internal logic of $\mathbf{C}$ would ordinarily be straightforward. However, our formulation will have one further twist. Rather than using the natural numbers object $\mathbf{N}$ of $\mathbf{C}$ for the encoding, it turns out to be useful to instead use a natural numbers object in the category, $\mathbf{P}$, of predomains. Of course, if Axiom $\mathbf{N}$ holds then $\mathbf{N}$ is itself a natural numbers object in $\mathbf{P}$. But, for the sake of applications, see Section 15.2, it is better not to assume Axiom $\mathbf{N}$ in general.

First we have to show that the category $\mathbf{P}$ has a natural numbers object. By Theorem 1, every fibred endofunctor on $\mathbf{pP}$ has a bifree algebra. In particular $\mathbf{1} + (-)\colon \mathbf{pP} \to \mathbf{pP}$ has a bifree algebra $[0_c, s_c]\colon \mathbf{1} + \mathbf{N_c} \rightharpoonup \mathbf{N_c}$. But isomorphisms in $\mathbf{pP}$ are total, and coproducts in $\mathbf{P}$ and $\mathbf{pP}$ agree, thus $[0_c, s_c]\colon \mathbf{1} + \mathbf{N_c} \longrightarrow \mathbf{N_c}$ is a map in $\mathbf{P}$.

Proposition 12.1 $\mathbf{1} \xrightarrow{0_c} \mathbf{N_c} \xleftarrow{s_c} \mathbf{N_c}$ *is a natural numbers object in* $\mathbf{P}$.

PROOF. We show that $[0_c, s_c]\colon \mathbf{1} + \mathbf{N_c} \longrightarrow \mathbf{N_c}$ is an initial algebra for the endofunctor $\mathbf{1} + (-)\colon \mathbf{P} \to \mathbf{P}$. Let $f\colon \mathbf{1} + X \longrightarrow X$ be any algebra for $\mathbf{1} + (-)$. Then f is also an algebra for the functor $\mathbf{1} + (-)\colon \mathbf{pP} \to \mathbf{pP}$. So, by the initiality of the algebra $[0_c, s_c]$ in $\mathbf{pP}$, there exists a unique partial map $g\colon \mathbf{N_c} \rightharpoonup X$ such that $g \circ [0_c, s_c] = f \circ (\mathbf{1} + g)$. It suffices to show that g is total. Define $Z = \{n'\colon \mathbf{N_c} \mid g(n')\!\downarrow\}$. Then Z is well-complete because it is a Σ-subobject of the well-complete $\mathbf{N_c}$. As f and $[0_c, s_c]$ are total, one shows easily that $0_c \in Z$ and also $s_c(n') \in Z$ for all $n' \in Z$. Thus Z carries a subalgebra of $[0_c, s_c]\colon \mathbf{1} + \mathbf{N_c} \longrightarrow \mathbf{N_c}$. As $[0_c, s_c]$ is an initial algebra in $\mathbf{pP}$ and Z is a predomain, it follows that the inclusion $Z \rightarrowtail \mathbf{N_c}$ is an isomorphism in $\mathbf{pP}$ and hence in $\mathbf{P}$. Thus $Z = \mathbf{N_c}$, i.e. g is indeed total. $\Box$

We have that $\mathbf{P}$ is a cartesian-closed category with natural numbers object $\mathbf{N_c}$. It follows that we can represent every k-ary primitive recursive function by a morphism $(\mathbf{N_c})^k \longrightarrow \mathbf{N_c}$, see [20, Part III] for details. Similarly, any k-ary primitive recursive predicate is represented by a morphism $(\mathbf{N_c})^k \longrightarrow \mathbf{2}$. We use such operations and predicates freely, tagging them (e.g. $+_c$, $<_c$, etc.) to emphasise that they are associated with $\mathbf{N_c}$ rather than with $\mathbf{N}$.

We refer to $\mathbf{N_c}$ as the object of *computational natural numbers* in $\mathbf{C}$. One has

to take care when reasoning about elements of $\mathbf{N_c}$ internally in $\mathbf{C}$ as induction is not always valid. (The validity of full induction is equivalent to Axiom **N**.) Nevertheless, as $\mathbf{N_c}$ is a natural numbers object in $\mathbf{P}$ it holds that, for any well-complete subobject $Z \rightarrowtail \mathbf{N_c}$ we have

$$\mathbf{C} \models 0_c \in Z \wedge (\forall n{:}\mathbf{N_c}.\, n \in Z \rightarrow s_c(n) \in Z) \rightarrow \forall n{:}\mathbf{N_c}.\, n \in Z\,, \qquad (41)$$

i.e. induction holds for well-complete predicates. By Axiom **2**, we have, in particular, that induction holds for logically decidable predicates. Combined with the previous discussion on primitive recursive functions, one therefore obtains that $\mathbf{N_c}$ is an internal model of *Intuitionistic Primitive Recursive Arithmetic (IPRA)* in $\mathbf{C}$.[4] In what follows, we shall use IPRA very informally, making only the occasional remark when further justification of an argument seems useful. The reader is referred to [12] for a rigorous definition of (classical) primitive recursive arithmetic and a thorough exploration of its properties.

Having IPRA at our disposal, the task of Gödel numbering, using $\mathbf{N_c}$ as the object of natural numbers, is routine. The precise choice of encoding is unimportant. What does matter is that types and terms are represented in a natural way using primitive recursive operations. We write $\mathcal{T}_\sigma \rightarrowtail \mathbf{N_c}$ and $\mathcal{V}_\sigma \rightarrowtail \mathbf{N_c}$ for the subobjects of (Gödel numbers of) closed terms and values of type σ respectively. The encoding should be chosen so that these are primitive recursive subobjects. For the operational semantics, we encode the proof system of Fig. 2 in such a way that the relation "Π is a derivation of $t \rightsquigarrow v$", for which we write $\Pi \vdash t \rightsquigarrow v$, is a primitive recursive ternary relation on Gödel numbers. In the internal logic of $\mathbf{C}$, we write: $t \rightsquigarrow v$ for $\exists\, \Pi{:}\mathbf{N_c}.\, (\Pi \vdash t \rightsquigarrow v)$; and $t\!\Downarrow$ for $\exists\, v{:}\mathbf{N_c}.\, (t \rightsquigarrow v)$. For convenience, we are here using the same notation for the formalized relations, expressed using Gödel numbers, as for the original external relations on terms. It will always be clear from the context which relation is meant. E.g. whenever an operational relation is expressed in the internal logic of $\mathbf{C}$, as in Proposition 12.2(2) below, the formalized relation on Gödel numbers is the one intended. We remark that the basic properties of the operational semantics are all provable internally $\mathbf{C}$ using only the methods of IPRA. Such results include: if $t{:}\,\sigma$ and $t \rightsquigarrow v$ then $v{:}\,\sigma$; and: if $t \rightsquigarrow v$ and $t \rightsquigarrow v'$ then $v = v'$. In both cases these are proved by quantifier-free inductions over primitive recursive predicates obtained by universally abstracting the existentially quantified variables in the hypotheses of the implications. We shall freely use such results about the formalized syntax without further comment.

The main result of this section establishes the equivalence of operational and denotational notions of convergence, as interpreted within $\mathbf{C}$. For the denotational notion, given a closed term $t{:}\,\sigma$, we write $(\!|t|\!)\downarrow$ for the internal statement in $\mathbf{C}$ that $(\!|t|\!){:}\,\mathbf{1} \rightharpoonup (\!|\sigma|\!)$ is a total function.

[4] Actually much stronger properties hold of $\mathbf{N_c}$. But being a model of IPRA is sufficient for our purposes.

Proposition 12.2 (Internal computational adequacy) *For closed* $t\colon \sigma$*:*

(1) $t\Downarrow$ *implies* $\mathbf{C} \models (\!|t|\!)\downarrow$.
(2) $\mathbf{C} \models (\!|t|\!)\downarrow \to t\Downarrow$.

Proposition 12.2(1) is proved by induction on derivations of the evaluation relation for t. Specifically, one proves that $t \rightsquigarrow v$ implies $\mathbf{C} \models (\!|t|\!) = (\!|v|\!)$. As it is easily shown that $\mathbf{C} \models (\!|v|\!)\downarrow$, for all values v, the result follows. See [3, Appendix C], for such an argument in detail. The interesting point concerning Proposition 12.2(1) is that it is not straightforward to internalize the above argument to obtain $\mathbf{C} \models\ t \Downarrow \to\ (\!|t|\!)\downarrow$. The fundamental obstacle here is that $(\!|t|\!)$ is defined by an external induction on the structure of terms t, so it is apparently not possible to formulate an induction hypotheses that can be established by an internal induction on derivations of $t \rightsquigarrow v$.

For the proof of Proposition 12.2(2), we adapt the approach of [33,3] to our setting. The strategy is to define binary relations relating closed terms to their internal denotations. A closed term $t\colon \sigma$ has a denotation $(\!|t|\!)\colon \mathsf{pP}(1, (\!|\sigma|\!))$. However, values $v\colon \sigma$ enjoy the extra property that $(\!|v|\!)\downarrow$, i.e. that $(\!|v|\!)\colon \mathsf{P}(1, (\!|\sigma|\!))$, using the hom-set inclusion given by

$$\mathsf{P}(A, B) \;=\; \{f\colon \mathsf{pP}(A, B) \mid f \text{ is total}\} \;\rightarrowtail\; \mathsf{pP}(A, B)\,,$$

which holds for any $A, B\colon |\mathsf{P}|$ (equivalently $A, B\colon |\mathsf{pP}|$). For each closed type σ, we define a binary relation in $\mathbf{C}$,

$$\preceq_\sigma \;\rightarrowtail\; \mathsf{P}(1, (\!|\sigma|\!)) \times \mathcal{V}_\sigma.$$

Moreover, given any relation $\preceq \;\rightarrowtail \mathsf{P}(1, A) \times \mathcal{V}_\sigma$, where $A\colon |\mathsf{P}|$, we define an associated relation $\precsim \;\rightarrowtail \mathsf{pP}(1, A) \times \mathcal{T}_\sigma$ by:

$$e \precsim t \;\text{ iff }\; e\downarrow \text{ implies } \exists v\colon \mathcal{V}_\sigma.\ t \rightsquigarrow v \text{ and } e \preceq v\,, \tag{42}$$

making use of the operational semantics as formalized in $\mathbf{C}$. Thus, in particular, each relation $\preceq_\sigma$ above determines an associated

$$\precsim_\sigma \;\rightarrowtail\; \mathsf{pP}(1, (\!|\sigma|\!)) \times \mathcal{T}_\sigma\,.$$

The relations $\preceq_\sigma$ are defined so that the following equivalences hold in $\mathbf{C}$.

$$d \preceq_{\sigma+\tau} \mathtt{inl}(v) \;\text{ iff }\; d = \mathrm{inl}(c) \text{ where } c \preceq_\sigma v \tag{43}$$
$$d \preceq_{\sigma+\tau} \mathtt{inr}(v) \;\text{ iff }\; d = \mathrm{inr}(c) \text{ where } c \preceq_\tau v \tag{44}$$
$$(c, d) \preceq_{\sigma\times\tau} (u, v) \;\text{ iff }\; c \preceq_\sigma u \text{ and } d \preceq_\tau v \tag{45}$$
$$f \preceq_{\sigma\to\tau} \lambda x.\, t \;\text{ iff }\; \forall\, d\colon \mathsf{P}(1, (\!|\sigma|\!)),\ \forall\, v\colon \mathcal{T}_\sigma.\ d \preceq_\sigma v \;\to\; f(d) \precsim_\tau t[v/x] \tag{46}$$
$$d \preceq_{\mu X.\sigma} \mathtt{intro}(v) \;\text{ iff }\; \epsilon_{\mu X.\sigma}(d) \preceq_{\sigma[\mu X.\sigma/X]} v \tag{47}$$

Here, the clause for $\mu X.\sigma$ involves the isomorphism (39), which, because it is an isomorphism in pP, is also an isomorphism in P.

Once relations have been defined satisfying the equivalences above, the lemma below can be established.

Lemma 12.3 *If* $x_1 : \tau_1, \ldots, x_k : \tau_k \vdash t : \sigma$ *then*

$$\begin{aligned}\mathbf{C} \models \forall\, d_1 : \mathsf{P}(1, (\!|\tau_1|\!)),\ \ldots\ , d_k : \mathsf{P}(1, (\!|\tau_k|\!)),\ \forall\, v_1 : \mathcal{V}_{\tau_1},\ \ldots\ , v_k : \mathcal{V}_{\tau_k}.\\ d_1 \preceq_{\tau_1} v_1 \wedge \ldots \wedge d_k \preceq_{\tau_k} v_k \ \rightarrow\ (\!|t|\!)_\Gamma(\vec{d}) \precsim_\sigma t[\vec{v}/\vec{x}],\end{aligned}$$

where we write Γ *for* $x_1 : \tau_1, \ldots, x_k : \tau_k$ *and* $\vec{d}$ *for the vector* $d_1, \ldots, d_k$*, etc.*

A similar lemma is proved as [3, Lemma 9.2.18]. The only difference in our case is that a statement has to be established in the internal logic of $\mathbf{C}$.

PROOF. By induction on the structure of t. We consider a single case.

Suppose t is $(\mathtt{case}(s)\ \mathtt{of}\ y_1.t_1\ \mathtt{or}\ y_2.t_2)$, where we have $\Gamma \vdash s : \sigma_1 + \sigma_2$ and $\Gamma,\, y_1 : \sigma_1 \vdash t_1 : \sigma$ and $\Gamma,\, y_2 : \sigma_2 \vdash t_2 : \sigma$. Internally in $\mathbf{C}$, consider any $\vec{d}$ and $\vec{v}$ such that $d_i \preceq_{\tau_i} v_i$ for each i with $1 \leq i \leq k$. We must show that the relation $(\!|t|\!)_\Gamma(\vec{d}) \precsim_\sigma t[\vec{v}/\vec{x}]$ holds. Accordingly, suppose that $(\!|t|\!)_\Gamma(\vec{d})\downarrow$. Then, by the definition of $(\!|\mathtt{case}(s)\ \mathtt{of}\ x_1.t_1\ \mathtt{or}\ x_2.t_2|\!)_\Gamma(\vec{d})$ either: (i), it holds that $(\!|t|\!)_\Gamma(\vec{d}) = (\!|t_1|\!)_{\Gamma,\, y_1 : \sigma_1}(\vec{d}, c)$, where $(\!|s|\!)_\Gamma(\vec{d}) = \mathrm{inl}(c)$; or, (ii), it holds that $(\!|t|\!)_\Gamma(\vec{d}) = (\!|t_2|\!)_{\Gamma,\, y_2 : \sigma_2}(\vec{d}, c)$, where $(\!|s|\!)_\Gamma(\vec{d}) = \mathrm{inr}(c)$.

In case (i), we must show that there exists $v : \mathcal{V}_\sigma$ such that $t[\vec{v}/\vec{x}] \rightsquigarrow v$ and $(\!|t_1|\!)_{\Gamma,\, y_1 : \sigma_1}(\vec{d}, c) \preceq_\sigma v$. However, $(\!|s|\!)_\Gamma(\vec{d}) = \mathrm{inl}(c)$, so, by the induction hypothesis for s, we have that $\mathrm{inl}(c) \precsim_{\sigma_1+\sigma_2} s[\vec{v}/\vec{x}]$. Thus there exists v' such that $s[\vec{v}/\vec{x}] \rightsquigarrow v'$ and $\mathrm{inl}(c) \preceq_{\sigma_1+\sigma_2} v'$. As v' is a value of type $\sigma_1 + \sigma_2$, it is of the form $\mathtt{inl}(v'')$ or $\mathtt{inr}(v'')$. As $\mathrm{inl}(c) \preceq_{\sigma_1+\sigma_2} v'$, we have, by (43) and (44), that v' is $\mathtt{inl}(v'')$ where $c \preceq_{\sigma_1} v''$. Thus, by the induction hypothesis for t_1, we have that $(\!|t_1|\!)_{\Gamma,\, : y_1\sigma_1}(\vec{d}, c) \precsim_\sigma t_1[\vec{v}, v'' \,/\, \vec{x}, y_1]$, i.e. $(\!|t|\!)_\Gamma(\vec{d}) \precsim_\sigma t_1[\vec{v}, v'' \,/\, \vec{x}, y_1]$. As $(\!|t|\!)_\Gamma(\vec{d})\downarrow$, there exists v such that $t_1[\vec{v}, v'' \,/\, \vec{x}, y_1] \rightsquigarrow v$ and $(\!|t|\!)_\Gamma(\vec{d}) \preceq_\sigma v$. Moreover, we have $s[\vec{v}/\vec{x}] \rightsquigarrow \mathtt{inl}(v'')$ and $t_1[\vec{v}, v'' \,/\, \vec{x}, y_1] \rightsquigarrow v$, so $(\mathtt{case}(s)\ \mathtt{of}\ y_1.t_1\ \mathtt{or}\ y_2.t_2)[\vec{v}/\vec{x}] \rightsquigarrow v$, i.e. $t[\vec{v}/\vec{x}] \rightsquigarrow v$. Thus v is the required element of $\mathcal{V}_\sigma$.

Case (ii) is dealt with in a similar way. $\Box$

Proposition 12.2(2) follows easily from Lemma 12.3. Take any closed term $t : \sigma$. Then, by Lemma 12.3, it holds that $\mathbf{C} \models (\!|t|\!) \precsim_\sigma t$. Hence, by (42), we have $\mathbf{C} \models (\!|t|\!)\downarrow \rightarrow (\exists v : \mathcal{V}_\sigma.\ t \rightsquigarrow v \wedge (\!|t|\!) \preceq_\sigma v)$. So indeed $\mathbf{C} \models (\!|t|\!)\downarrow \rightarrow t\Downarrow$.

It remains to define the $\preceq_\sigma$ relations. Because they are recursively specified, this takes a considerable amount of work. Although it seems possible to apply Pitts' method of defining relations [31], doing so would require the development of further machinery. Because we already have the technology of suitable categories at our disposal, it seems easier to adapt the techniques of [33,3].

For each closed σ, we define an internal category R_σ. Internally in $\mathbf{C}$, objects are pairs $R = (|R|, \preceq_R)$ satisfying:

(1) $|R|\colon \mathcal{P}_{\mathbf{S}}U$ is a predomain.
(2) $\preceq_R$ is a binary relation between $\mathsf{P}(1, |R|)$ and $\mathcal{V}_\sigma$.
(3) For all $v{:}\mathcal{V}_\sigma$, the set $\{d\colon \mathsf{P}(1, |R|) \mid d \preceq v\}$ is well-complete.

Thus $|\mathsf{R}_\sigma|$ is easily defined as a subobject of $\sum_{A:|\mathsf{P}|} \mathcal{P}_{\mathbf{S}}\,(\mathsf{P}(1, A) \times \mathcal{V}_\sigma)$ in $\mathbf{C}$. Internally in $\mathbf{C}$, a morphism $f\colon \mathsf{R}_\sigma(R, S)$ is given by a morphism $f\colon \mathsf{pP}(|R|, |S|)$ such that

$$\forall v\colon \mathcal{V}_\sigma.\ \forall d\colon \mathsf{P}(1, |R|).\ d \preceq_R v \ \rightarrow\ f(d) \precsim_S v\,,$$

using (42) to define $\precsim_S$ from $\preceq_S$. Thus $\mathsf{R}_\sigma(R, S)$ is defined as a subobject of $\mathsf{pP}(|R|, |S|)$. The identities and composition are inherited from pP. Note that there is an evident forgetful internal functor $U_\sigma\colon \mathsf{R}_\sigma \rightarrow \mathsf{pP}$.

The purpose of the R_σ categories is that we can use them to obtain a non-standard interpretation of types under which each closed type σ gets interpreted as an object $\{\!|\sigma|\!\}\colon |\mathsf{R}_\sigma|$. Thus $\{\!|\sigma|\!\}$ will be a pair $(A, \preceq)$. Moreover, the definition of $\{\!|\sigma|\!\}$ will ensure that $A = (\!|\sigma|\!)$, and that $\preceq$ is the required relation $\preceq_\sigma$. The method of defining the non-standard interpretation of types is similar to that used to obtain their ordinary (internal) interpretation.

Proposition 12.4 *For each closed σ, the internal category R_σ is suitable, and the forgetful $U_\sigma\colon \mathsf{R}_\sigma \rightarrow \mathsf{pP}$ is suitable functor.*

The proof of this proposition is postponed until Section 13.

Next we define useful functors on the R_σ categories, the first three of which act as relational "liftings" of the type constructors on pP. For closed σ, τ and $\mu X.\sigma'$, we define

$$\begin{aligned}
+_{\sigma,\tau}&\colon \mathsf{R}_\sigma \times \mathsf{R}_\tau \rightarrow \mathsf{R}_{\sigma+\tau}\\
\times_{\sigma,\tau}&\colon \mathsf{R}_\sigma \times \mathsf{R}_\tau \rightarrow \mathsf{R}_{\sigma\times\tau}\\
\rightharpoonup_{\sigma,\tau}&\colon \mathsf{R}_\sigma^{op} \times \mathsf{R}_\tau \rightarrow \mathsf{R}_{\sigma\rightarrow\tau}\,,\\
\mathbb{I}_{\mu X.\sigma'}&\colon \mathsf{R}_{\sigma'[\mu X.\sigma'/X]} \rightarrow \mathsf{R}_{\mu X.\sigma'}\,,
\end{aligned}$$

to have the following actions on objects,

$$\begin{aligned} |R +_{\sigma,\tau} S| &= |R| + |S| \\ |R \times_{\sigma,\tau} S| &= |R| \times |S| \\ |R \rightharpoonup_{\sigma,\tau} S| &= |R| \rightharpoonup |S| \\ |\mathbb{I}_{\mu X.\sigma'} R| &= |R| \end{aligned}$$

$$\begin{aligned} d \preceq_{(R +_{\sigma,\tau} S)} \mathtt{inl}(v) &\quad \text{iff} \quad d = \mathrm{inl}(c) \text{ where } c \preceq_R v \\ d \preceq_{(R +_{\sigma,\tau} S)} \mathtt{inr}(v) &\quad \text{iff} \quad d = \mathrm{inr}(c) \text{ where } c \preceq_S v \\ (c,d) \preceq_{(R \times_{\sigma,\tau} S)} (u,v) &\quad \text{iff} \quad c \preceq_R u \text{ and } d \preceq_S v \\ f \preceq_{(R \rightharpoonup_{\sigma,\tau} S)} \lambda x.\, t &\quad \text{iff} \quad \forall\, d \colon \mathsf{P}(1, |R|),\, v \colon \mathcal{T}_\sigma.\; d \preceq_R v \;\rightarrow\; f(d) \precsim_S t[v/x] \\ d \preceq_{(\mathbb{I}_{\mu X.\sigma} R)} \mathtt{intro}(v) &\quad \text{iff} \quad d \preceq_R v\,. \end{aligned}$$

Lemma 12.5 *For objects R, S and R' of R_σ, R_τ and $\mathsf{R}_{\sigma'[\mu X.\sigma'/X]}$ respectively, it is indeed the case that $R +_{\sigma,\tau} S$, $R \times_{\sigma,\tau} S$, $R \rightharpoonup_{\sigma,\tau} S$ and $\mathbb{I}_{\mu X.\sigma'} R'$ are objects of $\mathsf{R}_{\sigma+\tau}$, $\mathsf{R}_{\sigma\times\tau}$, $\mathsf{R}_{\sigma\to\tau}$ and $\mathsf{R}_{\mu X.\sigma'}$ respectively.*

The proof of this lemma is given in Section 13. Having now obtained the actions on objects, the corresponding actions on morphisms are easily defined.

To define the non-standard interpretation of types, we again interpret open types as functors. Moreover, because of the bivariance of $\rightharpoonup_{\sigma,\tau}$, we use the internal categories $\widehat{\mathsf{R}_\sigma} = \mathsf{R}_\sigma^{op} \times \mathsf{R}_\sigma$. By Propositions 12.4 and 7.2, $\widehat{\mathsf{R}_\sigma}$ is suitable. Moreover, the induced forgetful $\widehat{U_\sigma} \colon \widehat{\mathsf{R}_\sigma} \to \widehat{\mathsf{pP}}$ is a suitable functor.

Given an open type $\Theta \vdash \sigma$, where $\Theta = X_1, \ldots, X_k$, and closed types $\tau_1, \ldots, \tau_k$, we interpret σ relative to $\vec{\tau} = \tau_1, \ldots, \tau_k$ as a symmetric internal functor

$$\{\!|\Theta \vdash \sigma|\!\}_{\vec{\tau}} \colon\; \widehat{\mathsf{R}_{\tau_1}} \times \cdots \times \widehat{\mathsf{R}_{\tau_k}} \;\rightarrow\; \widehat{\mathsf{R}_{\sigma[\vec{\tau}/\Theta]}}\,,$$

where we write $\sigma[\vec{\tau}/\Theta]$ for $\sigma[\tau_1, \ldots, \tau_k/X_1, \ldots, X_k]$. The interpretation is defined by induction on the structure of σ. To give the definition, we write $\vec{R}$ for an object $((R_1^-, R_1^+), \ldots, (R_k^-, R_k^+))$ of $\widehat{\mathsf{R}_{\tau_1}} \times \cdots \times \widehat{\mathsf{R}_{\tau_k}}$.

$$\begin{aligned} \{\!|\Theta \vdash X_i|\!\}^+_{\vec{\tau}}\, \vec{R} &= R_i^+ \\ \{\!|\Theta \vdash \sigma_1 + \sigma_2|\!\}^+_{\vec{\tau}}\, \vec{R} &= \{\!|\Theta \vdash \sigma_1|\!\}^+_{\vec{\tau}}\, \vec{R} \;+_{\sigma_1[\vec{\tau}/\Theta],\, \sigma_2[\vec{\tau}/\Theta]}\; \{\!|\Theta \vdash \sigma_2|\!\}^+_{\vec{\tau}}\, \vec{R} \\ \{\!|\Theta \vdash \sigma_1 \times \sigma_2|\!\}^+_{\vec{\tau}}\, \vec{R} &= \{\!|\Theta \vdash \sigma_1|\!\}^+_{\vec{\tau}}\, \vec{R} \;\times_{\sigma_1[\vec{\tau}/\Theta],\, \sigma_2[\vec{\tau}/\Theta]}\; \{\!|\Theta \vdash \sigma_2|\!\}^+_{\vec{\tau}}\, \vec{R} \\ \{\!|\Theta \vdash \sigma_1 \to \sigma_2|\!\}^+_{\vec{\tau}}\, \vec{R} &= \{\!|\Theta \vdash \sigma_1|\!\}^-_{\vec{\tau}}\, \vec{R} \;\rightharpoonup_{\sigma_1[\vec{\tau}/\Theta],\, \sigma_2[\vec{\tau}/\Theta]}\; \{\!|\Theta \vdash \sigma_2|\!\}^+_{\vec{\tau}} \vec{R} \\ \{\!|\Theta \vdash \mu X.\sigma'|\!\}_{\vec{\tau}} &= (\,\widehat{\mathbb{I}_{\mu X.\sigma'}} \circ \{\!|\Theta, X \vdash \sigma'|\!\}_{\vec{\tau},\, \mu X.\sigma'}\,)^\dagger\,. \end{aligned}$$

The remarks made after the definition of the standard interpretation of types apply *mutatis mutandis* to the non-standard interpretation. Again, a substitution lemma holds.

Lemma 12.6 *For open types* $\Theta \vdash \sigma'_1, \ \ldots \ , \Theta \vdash \sigma'_l$ *and* $\Theta' \vdash \sigma$*, where* $\Theta = X_1, \ldots, X_k$ *and* $\Theta' = Y_1, \ldots, Y_l$*, and for closed types* $\tau_1, \ldots, \tau_k$*,*

$$\{\![\, \Theta \vdash \sigma[\vec{\sigma'}/\vec{Y}\,]\,]\!\}_{\vec{\tau}} \ = \ \{\![\, \Theta' \vdash \sigma \,]\!\}_{\sigma'_1[\vec{\tau}/\Theta],\ldots,\sigma'_l[\vec{\tau}/\Theta]} \circ (\{\![\Theta \vdash \sigma_1]\!\}_{\vec{\tau}}, \ldots, (\!|\Theta \vdash \sigma_l|\!)_{\vec{\tau}}) \,.$$

PROOF. As before, a straightforward induction on types, using Proposition 3.5 for recursive types. □

The next lemma states the relationship between the non-standard interpretation of types and the standard interpretation.

Lemma 12.7 *For any open type* $\Theta \vdash \sigma$*, where* $\Theta = X_1, \ldots, X_k$*, and closed types* $\tau_1, \ldots, \tau_k$*, the diagram below commutes.*

$$\begin{array}{ccc}
\widehat{\mathsf{R}_{\tau_1}} \times \cdots \times \widehat{\mathsf{R}_{\tau_k}} & \xrightarrow{\{\![\Theta \vdash \sigma]\!\}_{\vec{\tau}}} & \widehat{\mathsf{R}_{\sigma[\vec{\tau}/\Theta]}} \\
\Big\downarrow & & \Big\downarrow \\
\widehat{U_{\tau_1}} \times \cdots \times \widehat{U_{\tau_k}} & & \widehat{U_{\sigma[\vec{\tau}/\Theta]}} \\
\Big\downarrow & & \Big\downarrow \\
\widehat{\mathsf{pP}} \times \cdots \times \widehat{\mathsf{pP}} & \xrightarrow[(\!|\Theta \vdash \sigma|\!)]{} & \widehat{\mathsf{pP}} \,.
\end{array}$$

PROOF. By induction on the structure of σ. For non-recursive types the property is immediate from the definition of $\{\![\Theta \vdash \sigma]\!\}_{\vec{\tau}}$ in terms of functors that are relational liftings of the corresponding functors on pP. For a recursive type, $\Theta \vdash \mu X.\sigma$, the commutativity of the diagram is a direct application of Lemma 8.3, using the suitability of the functor $\widehat{U_{\mu X.\sigma[\vec{\tau}/\Theta]}}$. □

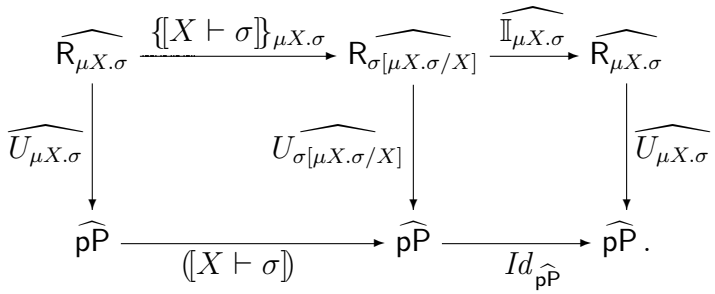
For closed types, the symmetric functor $\{\![\vdash \sigma]\!\} \colon \mathsf{1} \to \mathsf{R}_\sigma$, where $\mathsf{1}$ is the terminal internal category, corresponds to an object in R_σ of the form $(\{\![\sigma]\!\}, \{\![\sigma]\!\})$, where, by Lemma 12.7, $\{\![\sigma]\!\}$ is of the form $((\!|\sigma|\!), \preceq_\sigma)$. This, at last, defines the required relation $\preceq_\sigma$.

It remains to show that the $\preceq_\sigma$ relations satisfy the equivalences (43)–(47). For, (43)–(46), this is immediate from the definition of the $\{\![\Theta \vdash \sigma]\!\}_{\vec{\tau}}$ functors, as we have

$$\begin{array}{rcl}
((\!|\sigma + \tau|\!), \preceq_{\sigma+\tau}) & = & ((\!|\sigma|\!), \preceq_\sigma) \ +_{\sigma,\tau} \ ((\!|\tau|\!), \preceq_\tau) \\
((\!|\sigma \times \tau|\!), \preceq_{\sigma\times\tau}) & = & ((\!|\sigma|\!), \preceq_\sigma) \ \times_{\sigma,\tau} \ ((\!|\tau|\!), \preceq_\tau) \\
((\!|\sigma \rightharpoonup \tau|\!), \preceq_{\sigma\rightharpoonup\tau}) & = & ((\!|\sigma|\!), \preceq_\sigma) \ \rightharpoonup_{\sigma,\tau} \ ((\!|\tau|\!), \preceq_\tau) \,,
\end{array}$$

which is just a restatement of (43)–(46). Finally, for (47), we have that the object $(\{\![\mu X.\sigma]\!\}, \{\![\mu X.\sigma]\!\})$ of $\widehat{\mathsf{R}_{\mu X.\sigma}}$ carries the canonical bifree algebra for the

symmetric functor $\widehat{\mathbb{I}_{\mu X.\sigma}} \circ \widehat{\{\![X \vdash \sigma]\!\}_{\mu X.\sigma}} \colon \widehat{\mathsf{R}_{\mu X.\sigma}} \to \widehat{\mathsf{R}_{\mu X.\sigma}}$. By Lemma 12.7 and the definition of $\mathbb{I}_{\mu X.\sigma}$, the diagram below commutes.



Thus, by Lemma 8.2, the canonical bifree algebra for $\widehat{\mathbb{I}_{\mu X.\sigma}} \circ \widehat{\{\![X \vdash \sigma]\!\}_{\mu X.\sigma}}$ is:

$$(\epsilon_{\mu X.\sigma}, \iota_{\mu X.\sigma}) \colon \mathbb{I}_{\mu X.\sigma}\, \{\![X \vdash \sigma]\!\}_{\mu X.\sigma}\, (\{\![\mu X.\sigma]\!\}, \{\![\mu X.\sigma]\!\}) \longrightarrow (\{\![\mu X.\sigma]\!\}, \{\![\mu X.\sigma]\!\})\,,$$

in $\mathsf{R}_{\mu X.\sigma}$. By Lemma 12.6, this is

$$(\epsilon_{\mu X.\sigma}, \iota_{\mu X.\sigma}) \colon (\mathbb{I}_{\mu X.\sigma}\, \{\![\sigma[\mu X.\sigma/X]]\!\}, \mathbb{I}_{\mu X.\sigma}\, \{\![\sigma[\mu X.\sigma/X]]\!\}) \longrightarrow (\{\![\mu X.\sigma]\!\}, \{\![\mu X.\sigma]\!\})\,,$$

which unpacks to

$$\iota_{\mu X.\sigma} \colon \mathbb{I}_{\mu X.\sigma}\, \{\![\sigma[\mu X.\sigma/X]]\!\} \longrightarrow \{\![\mu X.\sigma]\!\} \tag{48}$$
$$\epsilon_{\mu X.\sigma} \colon \{\![\mu X.\sigma]\!\} \longrightarrow \mathbb{I}_{\mu X.\sigma}\, \{\![\sigma[\mu X.\sigma/X]]\!\}\,. \tag{49}$$

Take any $d \colon \mathsf{P}(\mathbf{1}, (\![\mu X.\sigma]\!))$. As $\iota_{\mu X.\sigma}$ and $\epsilon_{\mu X.\sigma}$ are mutual inverses, we have that $\epsilon_{\mu X.\sigma}(d)\!\downarrow$ and $\iota_{\mu X.\sigma}(\epsilon_{\mu X.\sigma}(d)) = d$. Thus, for any $v \colon \mathcal{V}_{\sigma[\mu X.\sigma/X]}$,

$$\begin{array}{lll} d \preceq_{\mu X.\sigma} \mathtt{intro}(v) & \text{iff } \epsilon_{\mu X.\sigma}(d) \preceq_{(\mathbb{I}_{\mu X.\sigma}\{\![\sigma[\mu X.\sigma/X]]\!\})} \mathtt{intro}(v) & \text{by (48) and (49)} \\ & \text{iff } \epsilon_{\mu X.\sigma}(d) \preceq_{\sigma[\mu X.\sigma/X]} v & \text{def. of } \mathbb{I}_{\mu X.\sigma}. \end{array}$$

Thus we have established (47). So the $\preceq_\sigma$ relations indeed satisfy all the required equivalences. This completes the proof of Proposition 12.2, modulo the proofs postponed to the next section.

13 Properties of R_σ

In this purely technical section, we give the promised proofs of Lemma 12.5 and Proposition 12.4.

First a necessary preliminary lemma, which should be compared to Proposition 2.5(10). The difference is that the lemma below is a consequence of Axiom **2**.

Lemma 13.1 $\mathbf{C} \models \forall P : \mathbf{2}^{\mathbf{N_c}}.\ (\exists n \colon \mathbf{N_c}.\ P(n)) \in \Sigma$.

PROOF. Define $d\colon \mathbf{2}^{\mathbf{N_c}} \longrightarrow \mathbf{1} + \mathbf{2}^{\mathbf{N_c}}$ by

$$d(P) \;=\; \begin{cases} \mathrm{inl}(*) & \text{if } P(0_c) \\ \mathrm{inr}(\lambda n\colon \mathbf{N_c}.\, P(s_c(n))) & \text{if } \neg(P(0_c)) \end{cases}$$

We have that $[0_c, s_c]\colon \mathbf{1} + \mathbf{N_c} \longrightarrow \mathbf{N_c}$ is a bifree algebra for the endofunctor $\mathbf{1} + (-)\colon \mathbf{pP} \to \mathbf{pP}$. Using the final coalgebra property of this, there exists a unique $h\colon \mathbf{2}^{\mathbf{N_c}} \longrightarrow \mathbf{N_c}$ in $\mathbf{pP}$ such that the diagram below commutes.

$$\begin{array}{ccc} \mathbf{1} + \mathbf{2}^{\mathbf{N_c}} & \xrightarrow{\;\mathsf{id} + h\;} & \mathbf{1} + \mathbf{N_c} \\ {\scriptstyle d}\uparrow & & \downarrow{\scriptstyle [0_c, s_c]} \\ \mathbf{2}^{\mathbf{N_c}} & \xrightarrow[\;h\;]{} & \mathbf{N_c} \end{array} \tag{50}$$

Our first goal is to establish that

$$\mathbf{C} \;\models\; \forall n\colon \mathbf{N_c}.\, \forall P\colon \mathbf{2}^{\mathbf{N_c}}.\, h(P) = n \;\leftrightarrow\; (P(n) \;\wedge\; \forall m <_c n.\, \neg P(m))\,, \tag{51}$$

where $h(P) = n$ is, of course, strict equality. This will be proved by induction on n. However, in order for such an induction to be justified, we must first show that the subobject

$$\{n\colon \mathbf{N_c} \mid \forall P\colon \mathbf{2}^{\mathbf{N_c}}.\, h(P) = n \;\leftrightarrow\; (P(n) \;\wedge\; \forall m <_c n.\, \neg P(m))\} \longrightarrow\!\!\!\!\!\rightarrow \mathbf{N_c} \tag{52}$$

is well-complete, see the remarks around (41).

To establish that (52) is well-complete it suffices, by Proposition 2.5(6), to show, for each $P\colon \mathbf{2}^{\mathbf{N_c}}$, that the subobjects

$$\{n\colon \mathbf{N_c} \mid h(P) = n \;\rightarrow\; (P(n) \;\wedge\; \forall m <_c n.\, \neg P(m))\} \longrightarrow\!\!\!\!\!\rightarrow \mathbf{N_c} \tag{53}$$

$$\{n\colon \mathbf{N_c} \mid (P(n) \;\wedge\; \forall m <_c n.\, \neg P(m)) \;\rightarrow\; h(P) = n\} \longrightarrow\!\!\!\!\!\rightarrow \mathbf{N_c} \tag{54}$$

are both well-complete. First, we consider (53). On the assumption that, $h(P)\!\downarrow$, we have $h(P)\colon \mathbf{N_c}$ and so

$$(h(P) = n \;\rightarrow\; (P(n) \;\wedge\; \forall m <_c n.\, \neg P(m))) \in \mathbf{2}\,,$$

whence (53) is a logically decidable subobject of a well-complete object, and hence well-complete. But

$$\begin{aligned} &\{n\colon \mathbf{N_c} \mid h(P) = n \;\rightarrow\; (P(n) \;\wedge\; \forall m <_c n.\, \neg P(m))\} \\ &\qquad = \bigcap_{x \in \{* \mid h(P)\downarrow\}} \{n\colon \mathbf{N_c} \mid h(P) = n \;\rightarrow\; (P(n) \;\wedge\; \forall m <_c n.\, \neg P(m))\}\,, \end{aligned}$$

where $\{* \mid h(P)\downarrow\}$ is the evident subobject of $\mathbf{1}$. We have shown that right-hand side is an intersection of well-complete subobjects of $\mathbf{N_c}$. Thus, by Proposition 2.5(6), we have that (53) is indeed well-complete. To show that (54) is well-complete, observe that $(P(n) \ \wedge \ \forall m <_c n.\, \neg P(m)) \in \mathbf{2}$ and $(h(P) = n) \in \Sigma$. However, given any $p : \mathbf{2}$ and $q : \Sigma$, we have $(p \to q) \in \Sigma$, by a trivial case analysis on p. Therefore,

$$((P(n) \ \wedge \ \forall m <_c n.\, \neg P(m)) \ \to \ h(P) = n) \in \Sigma\,.$$

Thus (54) is a Σ-subobject of a well-complete object, and hence well-complete. This completes the proof that (52) is well-complete.

We now prove (51) by induction on n. When $n = 0_c$, we have $h(P) = 0_c$ iff (by (50)) $d(P) = \mathrm{inl}(*)$, iff (by def. d) $P(0_c)$, iff $P(n) \wedge \forall m <_c n. \neg P(m)$. When $n = s_c(n')$, we have $h(P) = s_c(n')$ iff (by (50)) $d(P) = \mathrm{inr}(P')$, where $h(P') = n'$, iff (by def. d) $\neg(P(0_c))$ and $h(P') = n'$, where $P' = \lambda m : \mathbf{N_c}.\, P(s_c(m))$, iff (by induction hypothesis) $\neg(P(0_c))$ and $P'(n') \ \wedge \ \forall m <_c n'.\, \neg P'(m)$, iff (as $P'(m) = P(s_c(m))$) $P(n) \ \wedge \ \forall m <_c n.\, \neg P(m)$. This establishes (51).

We now use (51) to show that

$$\mathbf{C} \ \models \ (\exists n : \mathbf{N_c}.\ P(n)) \ \leftrightarrow \ h(P)\downarrow \ . \tag{55}$$

For the left-to-right implication, as P is a logically decidable predicate, we can prove by induction on $n : \mathbf{N_c}$ that

$$\mathbf{C} \ \models \ P(n) \ \to \ \exists m <_c n.\, (P(m) \ \wedge \ \forall m' <_c m.\, \neg P(m'))\,.$$

Now, reasoning in $\mathbf{C}$, assume $\exists n : \mathbf{N_c}.\ P(n)$. It follows from the implication above that $\exists n : \mathbf{N_c}.\ (P(n) \ \wedge \ \forall m <_c n.\, \neg P(m))$. Thus, by the right-to-left implication of (52), $\exists n : \mathbf{N_c}.\, h(P) = n$, i.e. $h(P)\downarrow$. For the converse implication of (55), suppose $h(P)\downarrow$. Then $h(P) = n$ for some n. So, by the left-to-right implication of (52), $P(n)$. Thus indeed $\exists n : \mathbf{N_c}.\ P(n)$.

We have established (55). The lemma follows, because $(h(P)\downarrow) \in \Sigma$. $\square$

For our purposes, the crucial consequence of Lemma 13.1 is that, for $t : \mathcal{T}_\sigma$ and $v : \mathcal{V}_\sigma$, we have $(t \rightsquigarrow v) \in \Sigma$ and $(t\Downarrow) \in \Sigma$. Indeed, the relation $\Pi \vdash t \rightsquigarrow v$ is a primitive recursive ternary relation, thus $(\Pi \vdash t \rightsquigarrow v) : \mathbf{2}$, for all $\Pi, t, v : \mathbf{N_c}$. So, by Lemma 13.1, $(\exists \Pi : \mathbf{N_c}.\, (\Pi \vdash t \rightsquigarrow v)) \in \Sigma$, i.e. $(t \rightsquigarrow v) \in \Sigma$. Also using a primitive recursive bijection $\mathbf{N_c} \times \mathbf{N_c} \cong \mathbf{N_c}$, it follows from Lemma 13.1 that $(\exists (\Pi, v) : \mathbf{N_c}.\, (\Pi \vdash t \rightsquigarrow v)) \in \Sigma$, i.e. $(t\Downarrow) \in \Sigma$.

We next introduce some convenient notation. Given an object $A : |\mathsf{P}|$, a relation $\preceq \ \rightarrowtail \mathsf{P}(1, A) \times \mathcal{V}_\sigma$, and $v : \mathcal{V}_\sigma$, define

$$A\restriction_{\preceq v} = \ \{d : \mathsf{P}(1, A) \mid d \preceq v\}\,.$$

Thus, $(A, \preceq)$ is an object of R_σ if and only if $A\restriction_{\preceq v}$ is well-complete for every v. Analogously, given $t : \mathcal{T}_\sigma$ define

$$A\restriction_{\precsim t} = \{e : \mathsf{pP}(1, A) \mid e \precsim t\}.$$

Lemma 13.2 *If $(A, \preceq)$ is an object of R_σ then, for any $t : \mathcal{T}_\sigma$, the subobject $A\restriction_{\precsim t}$ of A is well-complete.*

PROOF. To show that $A\restriction_{\precsim t}$ is well-complete, we apply Lemma 9.2, using the statement $t\Downarrow$ as the interpolant. By the remarks after Lemma 13.1, $t\Downarrow$ is indeed a Σ proposition.

To show that $A\restriction_{\precsim t}$ inhabited implies $t\Downarrow$, suppose that $A\restriction_{\precsim t}$ is inhabited. Thus there exists $e : \mathsf{pP}(1, A)$ such that $e \precsim t$. Then, by (42), there exists v such that $t \rightsquigarrow v$. Thus indeed $t\Downarrow$.

To show that $t\Downarrow$ implies $A\restriction_{\precsim t}$ is well-complete, suppose that $t\Downarrow$. Then there exists a unique v such that $t \rightsquigarrow v$. Therefore

$$\begin{aligned} A\restriction_{\precsim t} &= \{e : \mathsf{pP}(1, A) \mid e\downarrow \to e \preceq v\} && \text{by (42)} \\ &\cong \mathbb{L}\{d : \mathsf{P}(1, A) \mid d \preceq v\} \\ &= \mathbb{L}(A\restriction_{\preceq v}). \end{aligned}$$

Thus $A\restriction_{\precsim t}$ is well-complete by the closure of well-complete objects under $\mathbb{L}$.

We have established that the interpolation condition of Lemma 9.2 holds. Thus, by the lemma, $A\restriction_{\precsim t}$ is indeed well-complete. □

We now give the postponed proof of Lemma 12.5. Actually, in the case of $+_{\sigma,\tau}$, $\times_{\sigma,\tau}$ and $\mathbb{I}_{\mu X.\sigma'}$, the lemma is easily verified, so we just prove the $\rightharpoonup_{\sigma,\tau}$ case.

Lemma 13.3 *For objects R and S of R_σ and R_τ respectively, it holds that $R \rightharpoonup_{\sigma,\tau} S$ is an object of $\mathsf{R}_{\sigma\to\tau}$.*

PROOF. We must show that for $\lambda x.\, t : \mathcal{V}_{\sigma\to\tau}$ the set $(|R| \rightharpoonup |S|)\restriction_{\preceq_{R\rightharpoonup S}\, \lambda x.\, t}$ is well-complete. However,

$$(|R| \rightharpoonup |S|)\restriction_{\preceq_{R\rightharpoonup S}\, \lambda x.\, t} = \bigcap_{d \preceq_R v} \{f : \mathsf{P}(1, |R| \rightharpoonup |S|) \mid f(d) \precsim_S t[v/x]\}.$$

thus, by Proposition 2.5(6), it is enough to establish, for each $d : \mathsf{P}(1, |R|)$ and $v : \mathcal{V}_\sigma$ with $d \preceq_R v$, that the subobject $\{f : \mathsf{P}(1, |R| \rightharpoonup |S|) \mid f(d) \precsim_S t[v/x]\}$ is

well-complete. However, this subobject is obtained as $h^{-1}(|S| \restriction_{\precsim_S t[v/x]})$ where h is the map

$$\mathsf{P}(1, |R| \rightharpoonup |S|) \xrightarrow[f \mapsto f(d)]{} \mathsf{pP}(1, |S|)\,.$$

By Lemma 13.2, $|S| \restriction_{\precsim_S t[v/x]}$ is also well-complete. Thus indeed, by Proposition 2.5(7), $\{f : \mathsf{P}(1, |R| \rightharpoonup |S|) \mid f(d) \precsim_S t[v/x]\}$ is well-complete too. $\square$

Finally, we turn to the proof of Proposition 12.4. We must prove that R_σ is a suitable category, and that $U_\sigma \colon \mathsf{R}_\sigma \to \mathsf{pP}$ is a suitable functor.

Lemma 13.4 *The object $|\mathsf{R}_\sigma|$ carries a pointed structure $\bigvee : \mathbb{L}|\mathsf{R}_\sigma| \longrightarrow |\mathsf{R}_\sigma|$ such that the diagram below commutes.*

$$\begin{array}{ccc} \mathbb{L}|\mathsf{R}_\sigma| & \xrightarrow{\mathbb{L}\,U_\sigma} & \mathbb{L}|\mathsf{pP}| \\ \big\downarrow{\scriptstyle \bigvee} & & \big\downarrow{\scriptstyle \bigcup} \\ |\mathsf{R}_\sigma| & \xrightarrow{U_\sigma} & |\mathsf{pP}| \end{array} \tag{56}$$

PROOF. Formally $|\mathsf{R}_\sigma| = \sum_{A : |\mathsf{P}|} W_A$ where

$$W_A \;=\; \{\preceq : \mathcal{P}_{\mathbf{S}}(\mathsf{P}(1, A) \times \mathcal{V}_\sigma) \mid \text{for all } v{:}\mathcal{V}_\sigma,\ \ A \restriction_{\preceq v} \text{ is well-complete}\}\,.$$

Given any $A : |\mathsf{P}|$, we exhibit a pointed structure $w_A \colon \mathbb{L}(W_A) \longrightarrow W_A$.

For $e : \mathbb{L}(W_A)$ define $\preceq_e : \mathcal{P}_{\mathbf{S}}(\mathsf{P}(1, A) \times \mathcal{V}_\sigma)$ by

$$d \preceq_e v \quad \text{iff} \quad \text{for all } \preceq \in e, \text{ we have } d \preceq v\,.$$

Using Proposition 2.5(6), it is easily shown that $\preceq_e \in W_A$. We show that $w_A \colon e \mapsto \preceq_e$ satisfies the unit and multiplication laws, (6) and (7). The former is straightforward. For the latter, consider any $E : \mathbb{L}^2(W_A)$. We must show that $w_A(\bigcup E) = w_A(\{w_A(e) \mid e \in E\})$, i.e. that

$$d \preceq_{\bigcup E} v \quad \text{iff} \quad d \preceq_{\{\preceq_e \mid e \in E\}} v\,. \tag{57}$$

Accordingly, suppose that $d \preceq_{\bigcup E} v$. Then, for all $\preceq \in e \in E$ we have that $d \preceq v$. Consider any $\preceq \in \{\preceq_e \mid e \in E\}$. We must show that $d \preceq v$. But $\preceq = \preceq_e$ for some $e \in E$. We require that $d \preceq_e v$. But indeed, $d \preceq v$ for all $\preceq \in e$.

Conversely, suppose that $d \preceq_{\{\preceq_e \mid e \in E\}} v$. To show that $d \preceq_{\bigcup E} v$, consider any $\preceq \in e \in E$. We must show that $d \preceq v$. As $\preceq \in e \in E$, we have $e = \{\preceq\}$ and $E = \{e\}$. Thus $\{\preceq_{e'} \mid e' \in E\} = \{\preceq_e\} = \{\preceq\}$. But $d \preceq_{\{\preceq_{e'} \mid e' \in E\}} v$, i.e. $d \preceq_{\{\preceq\}} e$. Thus indeed $d \preceq v$. This establishes (57).

We have shown that each $A : |\mathsf{P}|$ carries a pointed structure $w_A : \mathbb{L}(W_A) \longrightarrow W_A$. By Lemma 9.3, $|\mathsf{P}| = |\mathsf{pP}|$ carries the pointed structure $\bigcup : \mathbb{L}|\mathsf{pP}| \longrightarrow |\mathsf{pP}|$. So, Lemma 4.10 now provides the required pointed structure $\bigvee : \mathbb{L}|\mathsf{R}_\sigma| \longrightarrow |\mathsf{R}_\sigma|$ making diagram (56) commute. □

Lemma 13.5 *For all $R, S : |\mathsf{R}_\sigma|$, it holds that $\mathsf{R}_\sigma(R, S)$ is complete.*

PROOF. We use Proposition 5.2.4 on $\mathsf{R}_\sigma(R, S)$ as a subobject of $\mathsf{pP}(|R|, |S|)$. Take any $f_{(-)} : (\mathsf{R}_\sigma(R, S))^{\mathbf{I}}$. By proposition 5.2.4, it suffices to show that, for all $d : \mathsf{P}(1, |R|)$ and $v : \mathcal{V}_\sigma$, it holds that $d \preceq_R v$ implies $(\bigsqcup_i f_i)(d) \precsim_S v$. Suppose then that $d \preceq_R v$. As $f_{(-)} : (\mathsf{R}_\sigma(R, S))^{\mathbf{I}}$, we have that $f_i(d) \precsim_S v$, for all $i : \mathbf{I}$. But then $f_{(-)}(d)$ is an $\mathbf{I}$-chain in $|S| \restriction_{\precsim_S v}$, which is, by Lemma 13.2, a complete subobject of $\mathsf{pP}(1, |S|)$. Thus, by the other direction of Proposition 5.2.4, $\bigsqcup_i(f_i(d)) \in |S| \restriction_{\precsim_S v}$, i.e. $\bigsqcup_i(f_i(d)) \precsim_S v$. However, $\bigsqcup_i(f_i(d)) = (\bigsqcup_i f_i)(d)$, by (24). Thus indeed $(\bigsqcup_i f_i)(d) \precsim_S v$. □

Lemma 13.6 *For all $R, S : |\mathsf{R}_\sigma|$, it holds that $\mathsf{R}_\sigma(R, S)$ is pointed, composition in R_σ is bistrict, and the forgetful $U_{RS} : \mathsf{R}_\sigma(R, S) \to \mathsf{pP}(|R|, |S|)$ is strict.*

PROOF. We show that the pointed structure on $\mathsf{pP}(|R|, |S|)$, see the explicit description after Lemma 9.4, restricts to a map $\beta'_{RS} : \mathbb{L}\,(\mathsf{R}_\sigma(R, S)) \longrightarrow \mathsf{R}_\sigma(R, S)$.

For $e : \mathbb{L}\,(\mathsf{R}_\sigma(R, S))$ define $\beta'_{RS}(e)$ to be the unique partial function $g : |R| \rightharpoonup |S|$ such that $g(x) = y$ if and only if there exists $f \in e$ such that $f(x) = y$. We must show that g is in $\mathsf{R}_\sigma(R, S)$, i.e. that $d \preceq_R v$ implies $g(d) \precsim_S v$. Suppose then that $d \preceq_R v$ and $g(d)\!\downarrow$. Then $e = \{f\}$ with $f : \mathsf{R}_\sigma(R, S)$ and $f(d)\!\downarrow$. But then $f(d) \preceq_S v$, i.e. $g(d) \preceq_S v$ as required.

As β'_{RS} is a restriction of an algebra for the $\mathbb{L}$-monad, it is itself an algebra for the monad. Thus $\mathsf{R}_\sigma(R, S)$ is indeed pointed. Moreover, by construction, the forgetful $U_{RS} : \mathsf{R}_\sigma(R, S) \to \mathsf{pP}(|R|, |S|)$ is strict. The bistrictness of composition is shown exactly as in Lemma 9.5. □

Lemma 13.7 $\{\mathsf{id}_R\}_{R : |\mathsf{R}_\sigma|}$ *is a strict family.*

PROOF. By Lemma 13.4, $|\mathsf{R}_\sigma|$ has pointed structure $\bigvee : \mathbb{L}|\mathsf{R}_\sigma| \longrightarrow |\mathsf{R}_\sigma|$ We must show equation (8), i.e. that $\mathsf{id}_{\bigvee e} = \beta'_{\bigvee e\, \bigvee^e}(\{\mathsf{id}_R \mid R \in e\})$, for all $e : \mathbb{L}|\mathsf{R}_\sigma|$. Take any $e : \mathbb{L}|\mathsf{R}_\sigma|$ and $x : |\bigvee e|$. Then $e = \{R\}$ for some R with $x : |R|$. So $\bigvee e = R$, whence $\beta'_{\bigvee e\, \bigvee^e}(\{\mathsf{id}_R \mid R \in e\})(x) = \mathsf{id}_R(x)$ as required. □

Lemma 13.8 *The functor $U_\sigma : \mathsf{R}_\sigma \to \mathsf{pP}$ creates bilimits of $\mathbf{I}$-bichains.*

PROOF. Let $(R_{(-)}, x_{(-)(-)})$ be an $\mathbf{I}$-bichain in R_σ. Then, writing $(A_i, \preceq_i)$ for R_i, we have that $(A_{(-)}, x_{(-)(-)})$ is an $\mathbf{I}$-bichain in pP. Let $(B, l_{(-)}, c_{(-)})$ be its bilimit, and let S be $(B, \preceq)$, where $\preceq \rightarrowtail \mathsf{P}(1, B) \times \mathcal{V}_\sigma$ is defined by

$$d \preceq v \quad \text{iff} \quad \forall i : \mathbf{I}.\ l_i(d) \precsim_i v\,.$$

We show below that $(S, l_{(-)}, c_{(-)})$ is a bilimit for $(R_{(-)}, x_{(-)(-)})$ in R_σ. The lemma then follows, as, by construction, $(S, l_{(-)}, c_{(-)})$ is created by U_σ.

First we show that S is indeed an object of R_σ. For this we need each $B \restriction_{\preceq v}$ to be well-complete. However, for each $v : \mathcal{V}_\sigma$,

$$B \restriction_{\preceq v} \;=\; \bigcap_{i:I} \{d : \mathsf{P}(1, B) \mid l_i(d) \precsim_i v\} \;=\; \bigcap_{i:I} l_i^{-1}\,(A_i \restriction_{\precsim_i v})\,.$$

Thus $B \restriction_{\preceq v}$ is indeed a predomain by Lemma 13.2 and Proposition 2.5(6–7).

To show that $(S, l_{(-)}, c_{(-)})$ is a bilimit in R_σ it suffices to show that each $l_i : \mathsf{pP}(B, A_i)$ indeed gives a morphism $l_i : \mathsf{R}_\sigma(S, R_i)$, and that each $c_i : \mathsf{pP}(A_i, B)$ indeed gives a morphism $c_i : \mathsf{R}_\sigma(R_i, S)$, because, if this is so, then the defining properties, (32) and (33), of a bilimit are inherited from pP. In the case of l_i, it is immediate from the definition of $\preceq$ that $l_i : \mathsf{R}_\sigma(S, R_i)$. To show $c_i : \mathsf{R}_\sigma(R_i, S)$, we must prove that, for any $d : \mathsf{P}(1, A_i)$ and $v : \mathcal{V}_\sigma$, it hold that $d \preceq_i v$ implies $c_i(d) \precsim v$. As $l_i \circ c_i = \mathsf{id}_{A_i}$, we have that $c_i(d)\!\downarrow$. So

$$\begin{array}{lll} c_i(d) \precsim v & \text{iff} \quad c_i(d) \preceq v & \\ & \text{iff} \quad \forall j : \mathbf{I}.\ l_j(c_i(d)) \precsim_j v & \text{def. of } \preceq \\ & \text{iff} \quad \forall j : \mathbf{I}.\ x_{ij}(d) \precsim_j v & \text{by (32) for } (B, l_{(-)}, c_{(-)}). \end{array}$$

However, $x_{ij} : \mathsf{R}_\sigma(R_i, R_j)$. Therefore $d \preceq_i v$ implies that $x_{ij}(d) \precsim_j v$, for all $j : \mathbf{I}$. Thus indeed, $d \preceq_i v$ implies $c_i(d) \precsim v$. $\square$

Taken together, Lemmas 13.4–13.8 provide a complete proof of Proposition 12.4.

14 External computational adequacy

Having proved internal computational adequacy, it is now possible to derive the second main result of the paper, a complete characterization of computational adequacy for the external interpretation of FPC, given in Section 11. Unsurpringly, the characterization depends upon relating properties of the computational natural numbers $\mathbf{N_c}$ in $\mathbf{C}$ to properties of the real world natural numbers $\mathbb{N}$.

We have remarked that, using the natural numbers object $\mathbf{N_c}$ of $\mathbf{P}$, one can define a standard encoding of any k-ary primitive recursive function as a morphism ${\mathbf{N_c}}^k \longrightarrow \mathbf{N_c}$, and of every k-ary primitive recursive predicate as a morphism ${\mathbf{N_c}}^k \longrightarrow \mathbf{2}$. In this section, it is necessary to be a little more precise about these encodings. Accordingly, we briefly review the details.

We think of a k-ary primitive recursive function $\phi_r\colon \mathbb{N}^k \to \mathbb{N}$, as being specified by "function letters" r in the term language of IPRA, see [12, §1.4]. This language simply provides a term structure suitable for generating the primitive recursive functions via projection, composition, primitive recursion, etc. For example, given terms r_1 and r_2 determining primitive recursive functions $\phi_{r_1}\colon \mathbb{N}^k \to \mathbb{N}$ and $\phi_{r_2}\colon \mathbb{N}^{k+2} \to \mathbb{N}$, we can form a term Rr_1r_2 representing the function $\phi_{Rr_1r_2}\colon \mathbb{N}^{k+1} \to \mathbb{N}$, defined from ϕ_{r_1} and ϕ_{r_2} by primitive recursion:

$$\begin{aligned} \phi_{Rr_1r_2}(i_1,\ldots,i_k,0) &= \phi_{r_1}(i_1,\ldots,i_k) \\ \phi_{Rr_1r_2}(i_1,\ldots,i_k,j+1) &= \phi_{r_2}(i_1,\ldots,i_k,j,\phi_{Rr_1r_2}(i_1,\ldots,i_k,j)) \end{aligned}$$

The interpretation of primitive recusive functions over $\mathbf{N_c}$ is defined by a straightforward induction on the structure of IPRA terms r. To each term r representing a k-ary function, we associate a morphism $f_r\colon {\mathbf{N_c}}^k \longrightarrow \mathbf{N_c}$. For example, the morphisms $f_{r_1}\colon {\mathbf{N_c}}^k \longrightarrow \mathbf{N_c}$ and $f_{r_2}\colon {\mathbf{N_c}}^{k+2} \longrightarrow \mathbf{N_c}$ determine $f_{Rr_1r_2}\colon {\mathbf{N_c}}^{k+1} \longrightarrow \mathbf{N_c}$ as the unique morphism fitting into the diagram below.

$$\begin{array}{ccc} {\mathbf{N_c}}^k + {\mathbf{N_c}}^k \times \mathbf{N_c} & \xrightarrow{\mathsf{id} + (\mathsf{id} \times \mathsf{id},\ f_{Rr_1r_2})} & {\mathbf{N_c}}^k + {\mathbf{N_c}}^k \times \mathbf{N_c} \times \mathbf{N_c} \\ \Big\downarrow{\scriptstyle [(\mathsf{id}, 0_c),\ (\mathsf{id} \times s_c)]} & & \Big\downarrow{\scriptstyle [f_{r_1}, f_{r_2}]} \\ {\mathbf{N_c}}^k \times \mathbf{N_c} & \xrightarrow[f_{Rr_1r_2}]{} & \mathbf{N_c} \end{array}$$

There is indeed a unique such morphism, because $\mathbf{N_c}$ is a natural numbers object in a cartesian closed category. The important property of the representation is that, for any r, and all $(i_1,\ldots,i_k) \in \mathbb{N}^k$,

$$f_r \circ (\overline{i_1},\ldots,\overline{i_k}) = \overline{\phi_r(i_1,\ldots,i_k)}, \tag{58}$$

where we write $\overline{(\cdot)}\colon \mathbb{N} \to \mathbf{C}(\mathbf{1}, \mathbf{N_c})$ for the evident function associating a "numeral" to each natural number.

Any term r representing a k-ary primitive recursive function, represents a k-ary primitive recursive predicate Φ_r, defined by $\Phi_r(i)$ iff $\phi_r(i) \neq 0$. By composing ϕ_r with the map $n \mapsto (n \neq 0_c)\colon \mathbf{N_c} \longrightarrow \mathbf{2}$, we represent Φ_r as a

morphism $P_r \colon \mathbf{N_c}^k \longrightarrow \mathbf{2}$. This satisfies, for all $(i_1, \ldots, i_k) \in \mathbb{N}^k$,

$$P_r \circ (\overline{i_1}, \ldots, \overline{i_k}, \overline{j}) \;=\; \begin{cases} \top & \text{if } \Phi_r(i_1, \ldots, i_k, j) \\ \bot & \text{if not } \Phi_r(i_1, \ldots, i_k, j) \end{cases} \tag{59}$$

Our characterization of computational adequacy involves the logical notion of 1-consistency, see e.g. [12, Def. 1.3.6], formulated for the computational natural numbers $\mathbf{N_c}$.

Definition 14.1 (Computational Σ^0_1-sentence) A *computational Σ^0_1-sentence* is a sentence, in the internal logic of $\mathbf{C}$, of the form $\exists n \colon \mathbf{N_c}.\, P_r(n)$, where r represents a unary primitive recursive predicate.

The adjective "computational" is included to emphasise that $\mathbf{N_c}$ is being used as the object of quantification rather than $\mathbf{N}$. If Axiom $\mathbf{N}$ holds then the computational Σ^0_1-sentences are just the ordinary Σ^0_1-sentences.

Definition 14.2 (Computational 1-consistency) We say that $\mathbf{C}$ is *computationally 1-consistent* if, for every computational Σ^0_1-sentence $\exists n \colon \mathbf{N_c}.\, P_r(n)$, $\mathbf{C} \models \exists n \colon \mathbf{N_c}.\, P_r(n)$ implies there exists $i \in \mathbb{N}$ such that $\Phi_r(i)$ holds.

In other words, $\mathbf{C}$ is computationally 1-consistent if the internal truth of computational Σ^0_1-sentences in $\mathbf{C}$ coincides with their external truth (n.b. it always holds, for such sentences, that external truth implies internal truth). The ordinary notion of 1-consistency, is defined in the same way, but using Σ^0_1-sentences over $\mathbf{N}$ in place of computational Σ^0_1-sentences. Using the canonical map $\mathbf{N} \longrightarrow \mathbf{N_c}$, it can be shown that, in general, computational 1-consistency implies ordinary 1-consistency. Thus computational 1-consistency is a stronger property than ordinary 1-consistency. However, when Axiom $\mathbf{N}$ holds, the two notions agree.

Theorem 2 (External computational adequacy) *The following are equivalent.*

(1) The interpretation of FPC *in* **pP** *is computationally adequate.*
(2) $\mathbf{C}$ *is computationally* 1*-consistent.*

PROOF. First we show that computational 1-consistency implies computational adequacy. By the definition of the external interpretation in terms of the internal one, it holds that $[\![t]\!]\downarrow$ if and only if $\mathbf{C} \models (\!|t|\!)\downarrow$. Therefore, it is immediate from Proposition 12.2(1) that $t \Downarrow$ implies $[\![t]\!]\downarrow$ To establish computational adequacy, we must show that also $[\![t]\!]\downarrow$ implies $t \Downarrow$. Accordingly, suppose that $[\![t]\!]\downarrow$. Then $\mathbf{C} \models (\!|t|\!)\downarrow$. So, by Proposition 12.2(2), $\mathbf{C} \models t \Downarrow$.

However, as in the remarks after Lemma 13.1, we have $t \Downarrow$ is of the form $\exists(\Pi, v) : \mathbf{N_c}.\ (\Pi \vdash t \rightsquigarrow v)$, which is a computational Σ^0_1-sentence. Thus, if $\mathbf{C}$ is computationally 1-consistent then indeed it holds that $t \Downarrow$.

To prove that computational adequacy implies computational 1-consistency, following [40, §6], we encode computational Σ^0_1-sentences as termination properties of FPC programs. First, we encode primitive recursive functions and predicates as FPC programs. This is routine, but we anyway outline the approach in order to justify that the relevant properties can be established of the encoding.

To each term r, representing a k-ary primitive recursive function, we associate a closed FPC term $t_r : \mathtt{nat}^k \to \mathtt{nat}$. The definition of t_r is by induction on the structure of r. For example, given FPC terms $t_{r_1} : \mathtt{nat}^k \to \mathtt{nat}$ and $t_{r_2} : \mathtt{nat}^{k+2} \to \mathtt{nat}$, then one defines a term $t_{Rr_1r_2} : \mathtt{nat}^{k+1} \to \mathtt{nat}$ by

$$\begin{aligned} t_{Rr_1r_2} \;=\; \mathtt{rec}\, f = & \lambda(x_1, \ldots, x_k, y).\ \mathtt{if}\ y = \overline{0}\ \mathtt{then}\ t_{r_1}(x_1, \ldots, x_k) \\ & \mathtt{else}\ t_{r_2}(x_1, \ldots, x_k, \mathtt{pred}(y), f(x_1, \ldots, x_k, \mathtt{pred}(y)))\,, \end{aligned}$$

using obvious abbreviations. The FPC terms t_r, defined in this way, represent the associated primitive recursive functions ϕ_r in the sense that, for all $i_1, \ldots, i_k \in \mathbb{N}$,

$$t_r(\overline{i_1}, \ldots, \overline{i_k}) \rightsquigarrow \overline{\phi_r(i_1, \ldots, i_k)}\,.$$

Using the evident isomorphism $\gamma : \mathbf{N_c} \longrightarrow \mathsf{P}(\mathsf{1}, (\!|\mathtt{nat}|\!))$, we claim that

$$\mathbf{C} \;\models\; \forall(n_1, \ldots, n_k) : {\mathbf{N_c}}^k.\ (\!|t_r|\!)(\gamma(n_1), \ldots, \gamma(n_k)) = \gamma(f_r(n_1, \ldots, n_k))\,, \quad (60)$$

where $f_r : {\mathbf{N_c}}^k \longrightarrow \mathbf{N_c}$ is as above. Property (60) is proved by induction on the structure of r. For example, the case for primitive recursion uses the fact that $(\!|\mathtt{rec}\ f = \lambda x.\, t|\!) = (\!|\lambda x.\, t[\mathtt{rec}\ f = \lambda x.\, t/x]|\!)$, which is easily shown. This implies that $(\!|t_{Rr_1r_2}|\!)$ gives rise to a partial map fitting into the defining diagram for $f_{Rr_1r_2}$ (modulo γ). However, the total map $f_{Rr_1r_2}$ is itself the only partial map fitting into this diagram. Thus $(\!|t_{Rr_1r_2}|\!)$ and $f_{Rr_1r_2}$ coincide.

Next, consider any primitive recursive (unary) predicate P_r. Using the encoding of primitive recursive functions above, one easily encodes P_r as a closed term $t'_r : \mathtt{nat} \to \mathtt{bool}$ satisfying:

$$t'_r(\overline{i}) \rightsquigarrow \mathtt{tt} \ \text{ if and only if } \ \Phi_r(i). \quad (61)$$

Also, by (60), t'_r satisfies

$$\mathbf{C} \;\models\; \forall n : \mathbf{N_c}.\ ((\!|t'_r|\!)(\gamma(n)) = (\!|\mathtt{tt}|\!)) \;\leftrightarrow\; P_r(n)\,. \quad (62)$$

We now encode the computational Σ^0_1-property $\exists n : \mathbf{N_c}.\, P_r(n)$ as a search pro-

gram in FPC. Consider the closed term $s_r : \mathtt{nat} \to \mathtt{unit}$ where s_r is:

$$\mathtt{rec}\, f = \lambda x.\, \mathtt{if}\ t'_r(x)\ \mathtt{then} * \mathtt{else}\ f(\mathtt{succ}(x))\,.$$

Intuitively, $s_r(\bar{i})$ searches for a number $j \geq i$ such that $\Phi_r(j)$, returning the canonical element $* : \mathtt{unit}$ if it finds such a j. By a straightforward induction on derivations in the operational semantics, it follows from (61) that:

$$\text{if } s_r(\overline{0})\Downarrow \text{ then there exists } i \text{ such that } \Phi_r(i). \tag{63}$$

We now show that also:

$$\mathbf{C} \;\models\; (\exists n : \mathbf{N_c}.\, P_r(n)) \;\to\; (\!|s_r(\overline{0})|\!)\downarrow\,. \tag{64}$$

To prove this, we show by internal induction on n that

$$\mathbf{C} \;\models\; \forall n : \mathbf{N_c}.\, \forall m : \mathbf{N_c}.\, P_r(n +_c m) \;\to\; (\!|s_r|\!)(\gamma(n))\downarrow\,, \tag{65}$$

from which (64) easily follows. By the discussion around (41), to justify the use of induction, we must show that

$$\{m : \mathbf{N_c} \mid \forall n : \mathbf{N_c}.\, P_r(n +_c m) \;\to\; (\!|s_r|\!)(\gamma(n))\downarrow\} \rightarrowtail \mathbf{N_c} \tag{66}$$

is well-complete. However, for each m, n, we have that $(P_r(n)) \in \mathbf{2}$ and $((\!|s_r|\!)(\gamma(n))\downarrow) \in \Sigma$, so $(P_r(n) \to (\!|s_r|\!)(\gamma(n))\downarrow) \in \Sigma$. Thus (66) is an intersection of well-complete subobjects, hence, by Proposition 2.5(6), well-complete. It remains to carry out the induction to establish (65). The argument is lengthy but routine. It uses (62), and also basic semantic equivalences, such as $(\!|\mathtt{rec}\, f = \lambda x.\, t|\!) = (\!|\lambda x.\, t[\mathtt{rec}\, f = \lambda x.\, t/x]|\!)$, and also, crucially, $(\!|s_r(\mathtt{succ}(x))|\!)_{x:\,\mathtt{nat}}(\gamma(n)) \simeq (\!|s_r|\!)(\gamma(n +_c 1))$, which is used in the induction step in order to apply the induction hypothesis.

Finally, we show that computational adequacy implies 1-consistency. Suppose that the interpretation is computationally adequate and that $\mathbf{C} \models \exists n : \mathbf{N_c}.\, P_r(n)$. Then, by (64), $\mathbf{C} \models (\!|s_r(\overline{0})|\!)\downarrow$, i.e. $[\![s_r(\overline{0})]\!]\downarrow$. Thus, by computational adequacy, $s_r(\overline{0})\Downarrow$. Therefore, by (63), there indeed exists i such that $\Phi_r(i)$. $\square$

We say that $\mathbf{C}$ is *trivial* if any of the following equivalent conditions holds: $\mathbf{C} \models \bot$, i.e. the internal logic is inconsistent; $\mathbf{0} \cong \mathbf{1}$; or $\mathbf{C}$ is equivalent to the terminal category. By the results of [41,43], the existence of a nontrivial $\mathbf{C}$ is equivalent to the consistency of IZF. Nontriviality implies many good properties of $\mathbf{C}$, e.g. all the computational numerals in $\mathbf{C}(\mathbf{1}, \mathbf{N_c})$ are distinct.

Obviously, computational adequacy is possible only when $\mathbf{C}$ is nontrivial. However, as in [40, Corollary 1], a consequence of Theorem 2 is that there exist nontrivial $\mathbf{C}$, satisfying Axiom **2** (or even Axiom **N**), for which the interpretation of FPC in **pP** is not computationally adequate. Indeed, this follows

from the completeness theorem for IZF with respect to categories with class structure [41,43], together with Gödel's incompleteness theorem for IZF with respect to true Π^0_1-sentences. Such categories $\mathbf{C}$ are pathologies. Instead, the main force of Theorem 2 is in the converse implication, which reduces computational adequacy to computational 1-consistency, which is a very weak condition. Indeed, we end the section with two results that are useful for demonstrating that computational 1-consistency holds for categories $\mathbf{C}$ that arise in practice.

We say that $\mathbf{N_c}$ is *standard* if the numeral map $\overline{(\cdot)}\colon \mathbb{N} \to \mathbf{C}(\mathbf{1}, \mathbf{N_c})$ is a bijection. If $\mathbf{N_c}$ is standard then $\mathbf{C}$ is clearly nontrivial.

Proposition 14.3 *If $\mathbf{N_c}$ is standard then $\mathbf{C}$ is computationally 1-consistent.*

PROOF. Suppose that $\mathbf{N_c}$ is standard, and suppose that $\mathbf{C} \models \exists n\colon \mathbf{N_c}.\, P_r(n)$. Using $h\colon \mathbf{2}^{\mathbf{N_c}} \longrightarrow\!\!\!\!\!\rightarrow \mathbf{N_c}$ defined in (50), we have, by (55), that $\mathbf{C} \models h(P_r)\downarrow$. Therefore the composite

$$h(P_r) \;=\; \mathbf{1} \xrightarrow{P_r} \mathbf{2}^{\mathbf{N_c}} \xrightarrow{\;h\;} \mathbf{N_c}$$

is total. Thus, by the standardness assumption $h(P_r) = \bar{i}\colon \mathbf{1} \longrightarrow \mathbf{N_c}$ for some $i \in \mathbb{N}$. Moreover, by (51), it holds that $\mathbf{C} \models P_r(h(P_r))$, i.e. that $\mathbf{C} \models P_r(\bar{i})$. Suppose, for contradiction, that $\Phi_r(i)$ is not true. Then, by (58), $\mathbf{C} \models \neg P_r(\bar{i})$, contradicting the consistency of $\mathbf{C}$. Thus indeed $\Phi_r(i)$ is true. $\Box$

Proposition 14.4 *If a countably infinite copower of $\mathbf{1}$ exists in $\mathbf{P}$ then $\mathbf{C}$ is computationally 1-consistent if it is nontrivial.*

PROOF. Suppose that $\mathbf{P}$ has a countably infinite copower $\coprod_{i\in\mathbb{N}} \mathbf{1}$. Then, straightforwardly, $\coprod_{i\in\mathbb{N}} \mathbf{1}$ is a natural numbers object, hence $\mathbf{N_c} \cong \coprod_{i\in\mathbb{N}} \mathbf{1}$. Thus $\mathbf{N_c}$ is a countable copower of $\mathbf{1}$. For each number $i \in \mathbb{N}$, we write $\bar{i}\colon \mathbf{1} \longrightarrow \mathbf{N_c}$ for the associated injection. Morover, we assign these injections in such a way that $\overline{0} = 0_c$ and such that $\overline{i+1} = s_c(\bar{i})$. (This is possible, by the proof that $\coprod_{i\in\mathbb{N}} \mathbf{1}$ is a natural numbers object.) Because $\mathbf{P}$ is cartesian closed, finite products distribute over arbitrary coproducts. Therefore $\{(\overline{i_1}, \ldots, \overline{i_k})\colon \mathbf{1} \longrightarrow {\mathbf{N_c}}^k\}_{(i_1,\ldots,i_k)\in\mathbb{N}^k}$ is a coproduct diagram in $\mathbf{P}$. By the coproduct property, for every function $\phi\colon \mathbb{N}^k \to \mathbb{N}$ there exists a unique morphism $g_\phi\colon {\mathbf{N_c}}^k \longrightarrow \mathbf{N_c}$ in $\mathbf{P}$ satisfying

$$g_\phi \circ (\overline{i_1}, \ldots, \overline{i_k}) \;=\; \overline{\phi(i_1, \ldots, i_k)}\,.$$

It is therefore immediate from (58) that $f_r = g_{\phi_r}$. Thus $\phi_r = \phi_{r'}$ implies $f_r = f_{r'}$. Similarly, it follows for (unary) primitive recursive predicates that

$P_r \colon \mathbf{N_c} \longrightarrow \mathbf{2}$ and $P_{r'} \colon \mathbf{N_c} \longrightarrow \mathbf{2}$ are equal whenever, for all $i \in \mathbb{N}$ it holds that $\Phi_r(i)$ iff $\Phi_{r'}(i)$.

We now establish computational 1-consistency. Suppose $\mathbf{C} \models \exists n \colon \mathbf{N_c}.\, P_r(n)$. Assume, for contradiction, that, for all i, it is not the case that $\Phi_r(i)$. Thus, as shown above, we have that $P_r \colon \mathbf{N_c} \longrightarrow \mathbf{2}$ is equal to the morphism $x \mapsto \bot$. But trivially $\mathbf{C} \models \neg\exists n \colon \mathbf{N_c}.\, \bot$, i.e. $\mathbf{C} \models \neg\exists n \colon \mathbf{N_c}.\, P_r(n)$, which implies the inconsistency of the internal logic. So, if $\mathbf{C}$ is nontrivial then there indeed exists i such that $\Phi_r(i)$. $\Box$

15 Applications

In this section, we give a brief outline of two applications of the results of this paper to derive computational adequacy for classes of concrete models of FPC. In contrast to previous approaches to computational adequacy for recursive types [33,3,25], the models we consider are not required to be order-enriched. Full details of the results outlined in this section will appear in [44].

Both applications follow the same general pattern. First, using standard techniques, we fully embed a concrete model of FPC into a topos $\mathbf{S}$. In order to apply the results of this paper, we further need the topos itself to arise as the full subcategory of small objects within a category $\mathbf{C}$ with class structure and universal object. The specific toposes we consider will either be Grothendieck toposes, see e.g. [24], or realizability toposes [13,15]. There are various results relating such toposes to categories with class structure in [19, Ch. IV], which can be massaged into an appropriate form to obtain embedding results sufficient for our needs. [5] However, it is cleaner to use a new and more general embedding theorem, which guarantees directly that every Grothendieck topos and every realizability topos arises as the full subcategory of small objects within a category with class structure and universal object. This result will appear in [2].

15.1 Realizability models

A realizability model is specified by a partial combinatory algebra $(A, \cdot)$, which determines a category $\mathbf{Mod}(A)$ of *modest sets* over A, see e.g. [22, §2–3]. In many such categories, one can find a dominance Σ, often conveniently determined by a *divergence* $D \subset A$ (see [22, Def. 4.1]), such that Axiom **2**

[5] This requires the application of [41, Theorem 7] to an initial ZF-algebra, in the sense of [19], constructed using an inaccessible cardinal. See [42] for further explanation.

holds. Numerous examples are presented in [21,22]. Furthermore, by [22, Theorem 7.5], Axiom **2** implies Axiom **N** in this setting.

As is well-known, there is a full embedding $\mathbf{Mod}(A) \hookrightarrow \mathbf{RT}(A)$ of modest sets into the *realizability topos* over A [13,15]. By the results of [2], we have, in turn, a full embedding $\mathbf{RT}(A) \hookrightarrow \mathbf{RC}(A)$, exhibiting $\mathbf{RT}(A)$ as the category of small objects in a category $\mathbf{RC}(A)$ with class structure and universal object. Then Σ is a dominance in $\mathbf{RC}(A)$ and $\mathbf{RC}(A)$ inherits Axiom **N** from $\mathbf{Mod}(A)$. Thus the results of this paper can be applied to obtain a category of predomains $\mathbf{P} \hookrightarrow \mathbf{RC}(A)$ in which FPC can be interpreted. Moreover, it can be shown that the interpretation of FPC lives within the subcategory $\mathbf{Mod}(A) \hookrightarrow \mathbf{RC}(A)$.

If A is nontrivial then the category $\mathbf{RC}(A)$ is computationally 1-consistent (equivalently 1-consistent) by Proposition 14.3, because it holds that the numerals $\mathbf{RC}(A)(\mathbf{1}, \mathbf{N}) \cong \mathbf{Mod}(A)(\mathbf{1}, \mathbf{N})$ are standard. Thus, by Theorem 2, the interpretation of FPC in $\mathbf{Mod}(A)$ is computationally adequate. This gives the first proof of computational adequacy for an interpretation of FPC in the realizability models of [11,28–30,21,22].

15.2 Models of axiomatic domain theory

In [3, Def. 8.3.1], an axiomatization of a general categorical notion of model for FPC is given. Moreover, as Theorem 9.2.19 of *op. cit.*, computational adequacy is proved for any nontrivial model satisfying two further conditions: (i) the model is *domain-theoretic*, i.e. it has an associated cpo-enrichment; and (ii) it is *absolute*, a condition which relates the partiality structure on the model to the cpo-enrichment. By applying the results of this paper, we obtain computational adequacy for a much wider class of models.

The class of models we work with is given by models of FPC, as in [3, Def. 8.3.1], that have an *inductive fixed-point object* in the sense of [5, Def. 1.11]. This class includes the domain-theoretic models of [4,3], and, more generally, all *KADT models*, as in [5, Def. 1.12]. Furthermore, in order to apply the Yoneda embedding below, we require our models to be small categories. (As usual, this can be circumvented for non-small categories, by moving to a larger set-theoretic universe.) Let $\mathcal{C}$ be any model in this class.

As $\mathcal{C}$ is a model of FPC, it has, in particular, a dominance Σ, an associated lifting functor $\mathbb{L}$, and finite coproducts. It can be proved that the latter are disjoint and stable (for disjointness see [3, Prop. 5.3.12]). The Yoneda functor thus gives a full embedding $\mathbf{y} \colon \mathcal{C} \hookrightarrow \mathbf{Sh}(\mathcal{C}, \mathsf{fc})$ where $\mathbf{Sh}(\mathcal{C}, \mathsf{fc})$ is the category of sheaves for the finite coproduct topology fc on $\mathcal{C}$. Moreover, it holds that $\mathbf{y}(\Sigma)$ is a dominance. Also, using the inductive fixed-point object in $\mathcal{C}$, it can

be proved that every object in the image of $\mathcal{C}$ under $\mathbf{y}$ is well-complete, cf. [5, Theorem 3.4.2]. As $\mathbf{y}$ preserves finite coproducts, it follows that Axiom **2** holds in $\mathbf{Sh}(\mathcal{C}, \mathsf{fc})$. (N.b. Axiom **N** need not hold in $\mathbf{Sh}(\mathcal{C}, \mathsf{fc})$, cf. [27, §4].) Also, the category of well-complete objects is a full reflective subcategory of $\mathbf{Sh}(\mathcal{C}, \mathsf{fc})$, cf. [5, Theorem 2.15].

By the results of [2], $\mathbf{Sh}(\mathcal{C}, \mathsf{fc})$ arises as the full subcategory of small objects in a category $\mathbf{ShC}(\mathcal{C}, \mathsf{fc})$ with class structure and universal object. Then $\mathbf{y}(\Sigma)$ is a dominance in $\mathbf{ShC}(\mathcal{C}, \mathsf{fc})$ and also $\mathbf{ShC}(\mathcal{C}, \mathsf{fc})$ inherits Axiom **2** from $\mathbf{Sh}(\mathcal{C}, \mathsf{fc})$. Thus the results of this paper can be applied to obtain a category of predomains $\mathbf{P} \hookrightarrow \mathbf{ShC}(\mathcal{C}, \mathsf{fc})$ in which FPC can be interpreted. Furthermore, it can be shown that the interpretation of FPC lives within the subcategory $\mathcal{C} \hookrightarrow \mathbf{ShC}(\mathcal{C}, \mathsf{fc})$.

Because well-complete objects form a full reflective subcategory of $\mathbf{Sh}(\mathcal{C}, \mathsf{fc})$, it holds that there exists a countable copower of $\mathbf{1}$ in $\mathbf{P}$. Hence, by Proposition 14.4 and Theorem 2, the interpretation of FPC in any nontrivial $\mathcal{C}$ is computationally adequate. As a special case, we obtain that the interpretation of FPC in any nontrivial domain-theoretic model (in the sense of [3, §8.5.1]) is computationally adequate. Thus it turns out that the absoluteness condition for computational adequacy in [4,3] is unnecessary. More generally, we have a computational adequacy result that applies to the wider classes of enriched models considered in [6,5], where the enrichment need not be an order enrichment. In particular, we obtain computational adequacy for all nontrivial KADT models in the sense of [5, Def. 1.12].

16 Discussion

In this paper we have, in a general axiomatic setting, proved algebraic compactness for the category of partial maps between small well-complete objects. Well-completeness is, however, just one of several proposed notions of predomain that have appeared in the synthetic domain theory literature, see e.g. [36,14,28,46,27] for other possibilities. It seems likely that our approach to algebraic compactness should easily adapt to such other notions of predomain. All that is required is to redo Section 9 in each case.

On the other hand, for the purposes of the applications of Section 15.2, it seems essential to use well-completeness as the notion of predomain. This is because, in general, models $\mathcal{C}$ of axiomatic domain theory do not appear to embed in any of the other (always smaller) categories of predomains within $\mathbf{Sh}(\mathcal{C}, \mathsf{fc})$. In particular, models of axiomatic domain theory are not necessarily order-enriched, and so cannot, in general, be embedded into any notion of predomain for which predomains are (either implicitly or explicitly) partially ordered.

Previous approaches to modelling recursive types within synthetic domain theory were either based on specific models [28], or instead restricted to axiomatic settings based on realizability models [40,35]. In the latter references, it is shown how the remarkable properties of small-complete small internal categories can be used to establish the algebraic compactness of the (Eilenberg-Moore) category of strict maps between pointed predomains. Such an approach does not adapt easily to establish algebraic compactness for the category of partial maps between predomains, because the category of partial maps is apparently not small-complete. Moreover, the approach is very much restricted to realizability models. (In sheaf models, for example, Freyd showed that the only small-complete small internal categories are preorders.) As the Eilenberg-Moore category is useful for modelling call-by-name and linear languages, it would be interesting to establish its algebraic compactness, in the general setting of this paper.

Acknowledgements

This paper derives from my 1993–5 collaboration with John Longley [22]. Throughout the long development of the work, I have benefited, in particular, from discussions with Marcelo Fiore, Pino Rosolini, Thomas Streicher and Paul Taylor, the last of whom is also acknowledged for his diagram macros.

References

[1] P. Aczel. The type theoretic interpretation of constructive set theory. In *Logic Colloquium '77*, North Holland, pages 55–66, 1978.

[2] S. Awodey, C. Butz, A.K. Simpson and Th. Streicher. Relating set theory, toposes and categories of classes. In preparation, 2003.

[3] M.P. Fiore. *Axiomatic Domain Theory in Categories of Partial Maps.* Distinguished Dissertation Series, CUP, 1996.

[4] M.P. Fiore and G.D. Plotkin. An Axiomatisation of computationally adequate domain-theoretic models of FPC. In *Proc. 9th IEEE Symposium on Logic in Computer Science*, pages 92-102, 1994.

[5] M.P. Fiore and G.D. Plotkin. An extension of models of axiomatic domain theory to models of synthetic domain theory. In *Proceedings of CSL 96*, pages 129–149. Springer LNCS 1258, 1997.

[6] M.P. Fiore, G.D. Plotkin, and A.J. Power. Complete cuboidal sets in axiomatic domain theory. In *Proc. 12th IEEE Symposium on Logic in Computer Science*, pages 268–279, 1997.

[7] M.P. Fiore and G. Rosolini. Domains in H. *Theoretical Computer Science*, 264:171–193, 2001.

[8] P.J. Freyd. Recursive types reduced to inductive types. In *Proc. 5th IEEE Symposium on Logic in Computer Science*, pages 498 – 507, 1990.

[9] P.J. Freyd. Algebraically complete categories. In *Category Theory, Proc. Como 1990*, Springer LNM 1488, pages 95–104, 1991.

[10] P.J. Freyd. Remarks on algebraically compact categories. In *Applications of Categories in Computer Science*, pages 95–106. LMS Lecture Notes 177, CUP, 1992.

[11] P.J. Freyd, P. Mulry, G. Rosolini, and D.S. Scott. Extensional PERs. *Information and Computation*, 98:211–227, 1992.

[12] J.-Y. Girard. *Proof Theory and Logical Complexity.* Bibliopolis, 1987.

[13] J.M.E. Hyland. The effective topos. In *L.E.J. Brouwer Centenary Symposium*, pages 165–216. North-Holland, 1982.

[14] J.M.E. Hyland. First steps in synthetic domain theory. In *Category Theory, Proc. Como 1990*, pages 131–156. Springer LNM 1488, 1991.

[15] J.M.E. Hyland, P.T. Johnstone, and A.M. Pitts. Tripos theory. *Math. Proc. Camb. Phil. Soc.*, 88:205–232, 1980.

[16] B. Jacobs. Semantics of weakening and contraction. *Ann. Pure Appl. Logic*, 69:73–106, 1994.

[17] B. Jacobs. *Categorical Logic and Type Theory.* North Holland, Amsterdam, 1999.

[18] M. Jibladze. A presentation of the initial lift algebra. *Journal of Pure and Applied Algebra*, 116:185–198, 1997.

[19] A. Joyal and I. Moerdijk. *Algebraic Set Theory.* LMS Lecture Notes 220, CUP, 1995.

[20] J. Lambek and P.J. Scott. *Introduction to Higher Order Categorical Logic.* Cambridge University Press, 1986.

[21] J.R. Longley. *Realizability Toposes and Language Semantics.* Ph.D. thesis, Department of Computer Science, University of Edinburgh. Available as ECS-LFCS-95-332, 1995.

[22] J.R. Longley and A.K. Simpson. A uniform account of domain theory in realizability models. *Math. Struct. in Comp. Sci.*, 7:469–505, 1997.

[23] Z. Luo. *Computation and Reasoning: A Type Theory for Computer Science.* Number 11 in International Series of Monographs on Computer Science. OUP, 1994.

[24] S. Mac Lane and I. Moerdijk. *Sheaves in Geometry and Logic: a First Introduction to Topos Theory.* Universitext. Springer Verlag, 1992.

[25] G. McCusker. *Games and Full Abstraction for a Functional Metalanguage with Recursive Types*. Distinguished Dissertation Series, Springer-Verlag, 1998.

[26] P. Martin-Löf. *Intuitionistic Type Theory*. Studies in Proof Theory. Bibliopolis, 1984.

[27] J. van Oosten and A.K. Simpson. Axioms and (counter)examples in synthetic domain theory. *Annals of Pure and Applied Logic*, 104:233–278, 2000.

[28] W.K.-S. Phoa. Effective domains and intrinsic structure. In *Proc. 5th IEEE Symposium on Logic in Computer Science*, pages 366–377, 1990.

[29] W.K.-S. Phoa. Building domains from graph models. *Math. Struct. in Comp. Sci.*, 2:277–299, 1992.

[30] W.K.-S. Phoa. From term models to domains. *Information and Computation*, 109:211–255, 1994.

[31] A.M. Pitts. Relational properties of domains. *Information and Computation*, 127:66–90, 1996.

[32] G.D. Plotkin. LCF considered as a programming language. *Theoretical Computer Science*, 5:223–255, 1977.

[33] G.D. Plotkin. Denotational semantics with partial functions. Lecture notes, C.S.L.I. Summer School, 1985.

[34] G.D. Plotkin. Algebraic compactness in an enriched stetting. Invited talk, Workshop on Logic, Domains and Programming Languages, Darmstadt, 1995.

[35] B. Reus and Th. Streicher. General synthetic domain theory — a logical approach. *Math. Struct. in Comp. Sci.*, 9:177–223, 1999.

[36] G. Rosolini. *Continuity and Effectivity in Topoi*. PhD thesis, University of Oxford, 1986.

[37] A. Ščedrov. Intuitionistic set theory. In *Harvey Friedman's Research on The Foundations of Mathematics*, pages 257–184. Elsevier Science Publishers, 1985.

[38] D.S. Scott. Relating theories of the λ-calculus. In *To H.B. Curry*, pages 403–450. Academic Press, 1980.

[39] A.K. Simpson. Recursive types in Kleisli categories. Unpublished manuscript, University of Edinburgh, 1992.

[40] A.K. Simpson. Computational adequacy in an elementary topos. In *Computer Science Logic, Proceedings CSL '98*, pages 323–342. Springer LNCS 1584, 1999.

[41] A.K. Simpson. Elementary axioms for categories of classes (extended abstract). In *Proc. 14th IEEE Symposium on Logic in Computer Science*, pages 77–85, 1999.

[42] A.K. Simpson. Computational Adequacy for Recursive Types in Models of Intuitionistic Set Theory. In *Proc. 17th IEEE Symposium on Logic in Computer Science*, 2002.

[43] A.K. Simpson. Elementary axioms for categories of classes. In preparation, 2003.

[44] A.K. Simpson. Computational adequacy for models of FPC. In preparation, 2003.

[45] M.B. Smyth and G.D. Plotkin. The category-theoretic solution of recursive domain equations. *SIAM Journal of Computing*, 11:761–783, 1982.

[46] P. Taylor. The fixed point property in synthetic domain theory. In *Proc. 6th IEEE Symposium on Logic in Computer Science*, pages 152–160, 1991.

[47] P. Taylor. *Practical Foundations*. Cambridge University Press, 1999.